
MAGAZINE
MSX
AÑO III
Núm. 21
Febrero
1987
300 Ptas.
Aplicaciones matemáticas con MSX
SVI 328: el port del cassette
Código máquina: rutina para buscar pokes
COMPATIBILIDAD ¿VERDAD O MENTIRA?

DIRECTOR:
Juan Arencibia.
COLABORADORES:
Angel Zarazaga, Teresa Aranda, Ricardo García.
DISEÑO:
Benito Gil.
Editada por:
PUBLINFORMATICA, S.A.
C/ Bravo Murillo, 377 - 5.° A
Tel.: 733 71 13
28020 Madrid.
Telex 488877 OPZXE
PRESIDENTE:
Fernando Bolín.
DIRECTOR EDITORIAL REVISTAS DE USUARIOS:
Juan Arencibia.
DIRECTOR DE VENTAS:
Antonio González.
JEFE DE PRODUCCION:
Miguel Onieva.
SERVICIO AL CLIENTE:
Julia González.
Tel.: 733 79 69
DIRECCION, REDACCION Y ADMINISTRACION:
C/ Bravo Murillo, 377 - 5.° A
Tel.: 733 74 13
28020 Madrid.
COORDINADORA DE PUBLICIDAD:
Silvia Bolín.
PUUBLICIDAD EN MADRID:
Emilio García.
PUBLICIDAD EN BARCELONA:
Lidia Cendros.
C/ Pelayo, 12.
Tel.: (93) 301 47 00 Ext. 27-28
08001 Barcelona.
Depósito Legal: M. 16.755-1985
Impreso en G. Velasco, S.A.
C/ Antonio Cabezón, 13. Madrid.
Distribuye:
S.G.E.L. Avda. Valdelaparra, s/n. Alcobendas (Madrid).
DISTRIBUIDORES:
VENEZUELA: SIPAM, S.A. Avda. República Dominicana, 541
ARGENTINA: DISTRIBUIDORA INTERCONTINENTAL BUENOS AIRES.

El P.V.P. para Ceuta, Melilla y Canarias, incluido servicio aéreo será de 300 ptas. sin I.V.A.

SUSCRIPCIONES:
Rogamos dirija toda la correspondencia relacionada con suscripciones a:
MSX
EDISA: Tel. 415 97 12
C/López de Hoyos, 141-5.°
28002 MADRID
(Para todos los pagos reseñar solamente MSX)
Para la compra de ejemplares atrasados diríjanse a la propia editorial
MSX
C/Bravo Murillo, 377-5.° A
Tel. 733 74 13 28020 MADRID

Si deseas colaborar en MSX remite tus artículos o programas a Bravo Murillo 377, 5.° A. 28020 Madrid. Los programas deberán estar grabados en cassette y los artículos mecanografiados.
A efectos de remuneración, se analiza cada colaboración aisladamente, estudiando su complejidad y calidad.



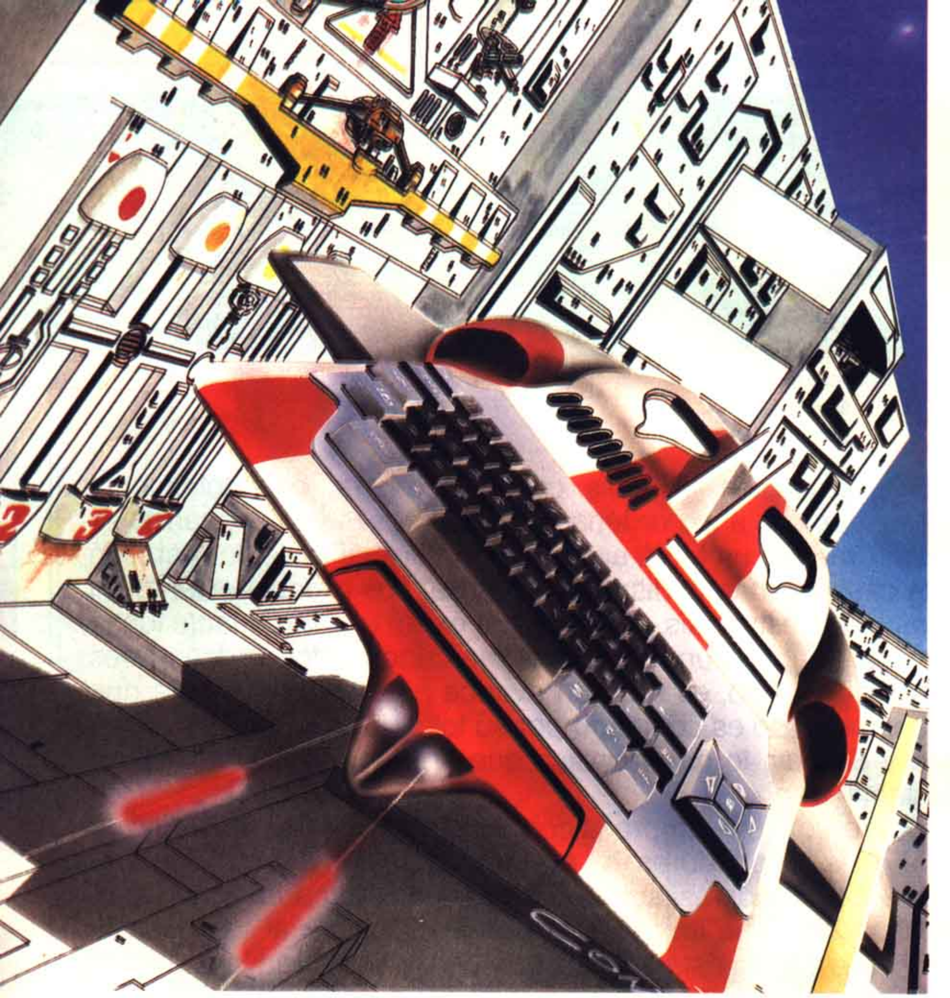
Cuando en un país como Inglaterra, casi totalmente cerrado a los ordenadores japoneses y más cuando se trata de un MSX, aparece una compañía de software, en este caso *Mirrorsoft,* dispuesta a lanzar sus productos tanto allí como en nuestro país, significa que existe una importante lucha por hacerse con una parte del mercado de los ordenadores domésticos. Esto, es un gran avance para el estándar, ya que aquel mercado es conocido por su total dedicación a los *Amstrad* y *Sinclair,* lo que significa que MSX sigue defendiéndose bastante bien.

Mirrorsoft, una empresa conocida en el ámbito del *software* para otros ordenadores, ha adquirido dos compañías americanas, fortaleciendo así su presencia en los mercados internacionales y dando un importante paso en el apoyo al estándar.

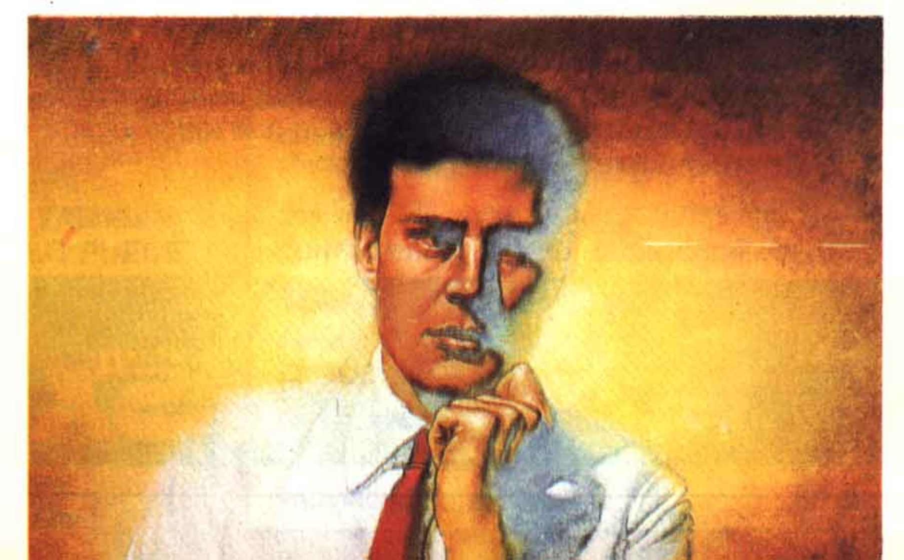
Por otro lado, este mes ha sido una gran noticia la aparición del *Spectravideo X'press 16.* Puede ser, como suele decirse, el arma definitiva tanto de esta empresa como para el mercado de los ordenadores personales. La batalla está por empezar, puesto que estamos ante un ordenador que, con un precio muy competitivo (109.500) pts.), es compatible PC y posee además, el chip de sonido y de gráficos de los ordenadores MSX de la II generación, con lo cual adquiere una potencia fuera de lo corriente. También cuenta con unas pequeñas sorpresas que contaremos en número venideros.

Por último, destacar que el tema de portada versa sobre la compatibilidad de la norma MSX. Es un tema, sobre el cual se han vertido auténticos ríos de tinta, sin haber entrado en la profundidad que este tema requiere. Hemos procurado llegar al fondo del problema y parece que no es una cuestión técnica la que provoca incompatibilidad entre los ordenadores. Va más allá, la incompatibilidad viene de no respetar lo que en principio era obligado, las características técnicas a la que todos y cada uno de los ordenadores están sujetos.










mario

NOTICIAS

El nuevo del Barrio; Mirrorsoft

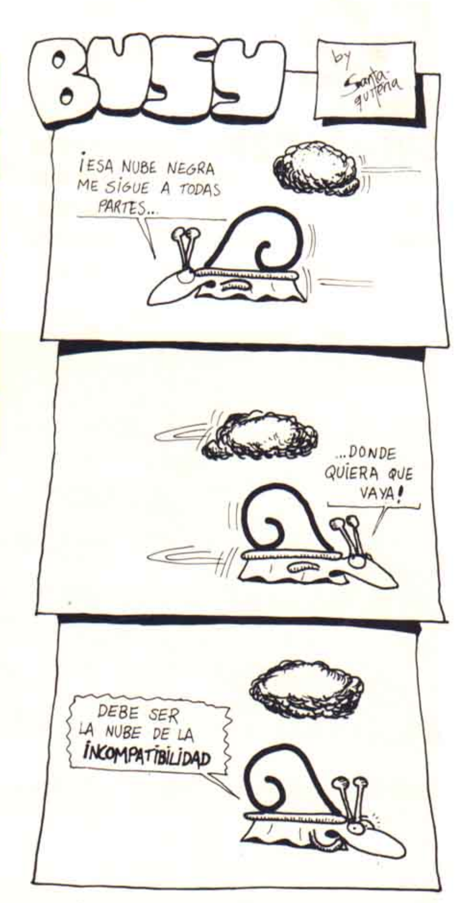
Esta compañía ha iniciado una ofensia en aquel país que, poco a poco, llegará hasta nosotros. Prueba de ello, son los primeros programas que comentamos en nuestras páginas de este mes (aunque *Dynamite Dan* ya es conocido). Además de introducirse en el mercado del *software* para MSX, también lo está haciendo en el mercado de las publicaciones y lo han hecho a lo grande, adquiriendo dos importantes compañías americanas como Spectrum Holobyte Inc. y Nexa Corporation.

Con este importante paso, se aseguran su presencia en el mercado internacional tanto de *software*, como editorial.

GOTO '87, Salón de la Informática, Ofimática y Equipos Afines

En Sevilla se ha celebrado, durante los días 27 a 31 de enero, GOTO '87, una feria organizada con la colaboración de la institución-feria de Muestras Iberoamericana de Sevilla. En ella estuvieron representados los sectores de ordenadores, periféricos, soportes, software, etc., siendo muy completo.

Durante la celebración de la feria, se desarrollaron unas jornadas técnicas, encaminadas a resolver los posibles problemas de los diversos sectores que concurrieron al salón.



Presentación del Spectravideo X'Press 16

El nuevo X'press 16 es un aparato encaminado a sustituir al MSX X'-press y dirigido hacia un mercado potencial bastante importante y numeroso. El nuevo X'press 16 es un ordenador compatible con el PC, con una memoria *RAM* de 256K, una unidad de disco de 5.25" incorporada, un conector de expansión, un teclado de 83 teclas y *ports* de expansión para *joysticks,* ratón, impresora paralela y salida de monitor RGB analógica, RGB digital y Video Compuesto.

Soporta ambos sistemas operativos PC-DOS y MS-DOS. En él pueden correr todos los paquetes de programas para PC, desde los conocidos *Wordstar* y *Multiplan,* así como los más recientes *Lotus 123, dBase III, Sidekick* y *Framework.*

El X'press 16 es el primer ordenador personal en incorporar dos interfaces gráficos; un adaptador de gráficos color compatible PC y el MVDP (el conocido procesador de video del MSX II).

La versión inicial del X'press 16 saldrá al mercado a un precio de 109.500 pts. (con una unidad de disco) y con la posibilidad de conectarlo a la TV, algo que ningún PC ni compatible puede hacer. La competitividad del precio se verá reflejada en el nivel de compatibilidad del ordenador.

Adaptador de MSX para SVI-328

El adaptador MSX para SVI-328 de C.C.G. está íntegramente diseñado y fabricado por el C.C.G. S&H, firma afincada en Bilbao y constituída por un grupo de programadores y técnicos residentes en la provincia de Vizcaya. El presente adaptador es el primer producto de fabricación propia que pone C.C.G. en el mercado español y está comercializándose desde primeros de noviembre del 86. La distribución se canaliza casi totalmente a través de la propia empresa, aunque existen establecimientos que lo distribuyen igualmente, siempre dependiendo del abastecimiento de C.C.G., en diversas provincias de la geografía española. El adaptador es físicamente un cartucho, que se conecta en el slot habitual de cartuchos del SVI-328/318, situado en la parte superior derecha del teclado. Por su diseño, puede dejarse conectado permanentemente o puede extraerse del slot cuando no se utilice.

Se trata de una adaptador para software que se encuentra en formato cassette o disco, y un paso de gigante, que permitirá unas innovaciones hasta la fecha desconocidas. Los usuarios del SVI-328 ya podrán ejecutar la mayoría de los programas existentes para MSX (como es lógico, siempre hay una excepción).

Para obtener más información, dirigirse a:

C.C.G. S&H
Larrasolo, 13, 4.° B
Baracaldo (48902 Vizcaya)

Caja de seguridad para discos

Advantek International Limited, de Hong Kong, ha lanzado al mercado una caja fuerte, con cerradura y a prueba de incendio, llamada *«Disksafe».* Está hecha con un metal que, además de proteger su contenido del fuego, reduce el riesgo de que el disco o cinta queden perjudicados por cualquier campo magnético presente.

Su diseño es bastante atractivo y no ocupa demasiado espacio, puesto que mide 210 x 190 x 350 cm. Su capacidad es de 80 discos de cualquier tamaño y aún queda espacio libre para cintas.

Otra novedad de esta empresa es el ordenador para los deportistas.

El *Sports Computer* es un aparato moderno, diseñado para las personas que quieren ponerse en forma. Se trata de una máquina muy sencilla que mide el ritmo de velocidad, proporciona un nuevo aliciente al deporte y vigila la salud del usuario controlando su pulso. Se han diseñado varias versiones del *Sports Computer,* para varios deportes específicos, tales como bicicleta fija, remo, jogging y esquí y puede adaptarse a cualquier tipo de actividad. Hay varios modelos de *Sports Computer* en el mercado, poseyendo esencialmente las mismas características. Para obtener más información, dirigirse a:

Hong Kong Trade Developement Council
Balmes, 184
08006 Barcelona



Compatibilidad:
¿verdad



o mentira?

De vez en cuando aparecen en algunas publicaciones ciertos comentarios que ponen en duda lo que se supone es la base de todos los equipos MSX: la COMPATIBILIDAD. Estos comentarios se suelen fundar en tres apariencias: programas que cargan en unas máquinas sí y en otras no, programas que fallan al conectar la unidad de disco y diferencia en los formatos de los *diskettes.* Ultimamente aparece otro refiriéndose a los programas de la Primera Generación que no funcionan en ordenadores de la Segunda, cuando en teoría debía hacerlo.

El primer fallo suele ser el más evidente. Compramos un programa, lo cargamos en nuestro ordenador X y funciona sin ningún problema. Lo llevamos a casa de un amigo que tiene también un ordenador MSX, pero de marca Y, tecleamos la correspondiente orden de carga y comprobamos con desesperación que, una vez llegado el final de la cinta, el ordenador queda totalmente insensible. Resultado; después de pulsar frustrados el *«RESET»* llegamos a la convicción de que, cuando nos hablaron de la famosa compatibilidad nos contaron un «camelo».

¿Realmente es así? Nada más lejos de la realidad. Ni el fabricante del ordenador X ni el del ordenador Y nos han engañado. El programa no funciona porque el programador que lo diseño no tuvo en cuenta que ciertas características internas de estos aparatos no tienen forzosamente que ser iguales en todos ellos. Este problema no hubiese aparecido si el programador hubiese tenido en cuenta que MSX no es una máquina en particular, sino un Sistema Operativo que varios fabricantes han decidido adoptar. Este Sistema Operativo ofrece la gran ventaja de hacer que todo programa que respete una serie de normas funcione perfectamente en cualquier máquina que lo use.

Para aclarar ideas vamos a estudiar algunos conceptos sobre lo que es un Sistema Operativo, su necesidad y lo que es un Programa Transportable, o sea, capaz de funcionar en máquinas físicamente diferentes pero que usan el mismo Sistema Operativo.

Necesidad de los Sistemas Operativos

Ningún ordenador puede funcionar sin la existencia de un medio que permita controlar el tráfico de datos entre la unidad central de proceso (CPU) y sus distintos periféricos (memoria, teclado, pantalla...). Este control se consigue con un programa que depende totalmente del microprocesador que hace de CPU y del diseño electrónico del ordenador, y suele llamarse Monitor o *BIOS (Basic Input Output System* = Sistema Básico de Entrada y Salida). Este programa necesita acompañarse de una extensión que permita al usuario introducir un mínimo de órdenes que le permitan efectuar algunas tareas básicas, como cargar otros programas o manipular ficheros (borrar, copiar...); esta extensión es el Procesador de Comandos. A este conjunto se le llama Sistema Operativo.

El Sistema Operativo libera, por tanto, al usuario de la necesidad de conocer y controlar los procesos internos del ordenador y le permite fijar su atención en sus propios problemas. Podemos compararlo a la corriente eléctrica con que funcionan nuestros electrodomésticos; sin ella no los podemos usar, pero no es necesario que sepamos cómo se produce o actúa para servirnos de sus posibilidades.

Evolución de los Sistemas Operativos

En principio, estos programas eran un conglomerado de rutinas puestas una a continuación de otra, sin más estructura que la necesaria para funcionar en el ordenador para el que habían sido diseñados. A cada rutina se accedía directamente, con lo que el conjunto resultaba muy rígido ya que, si la rutina se modificaba por tener un fallo o por haberla tenido que adaptar a un cambio en el ordenador, esa llamada podía dejar de tener validez, con lo que cualquier programa que la usase ya no podría funcionar sin sufrir una modificación. El resultado es que se perdía muchísimo tiempo adaptando cada programa a diferentes máquinas.

Otro fallo consistía en no separar el Procesador de Comandos del *BIOS* propiamente dicho. Ya que aquél sólo acepta un pequeño conjunto de órdenes, la gran mayoría de los programas tienen sus propios medios de comunicarse con el usuario haciéndolo innecesario.

Si el *BIOS* y el Procesador de Comandos no están claramente separados será muy difícil eliminar éste y aprovechar el espacio que ocupa.

Los Sistemas Operativos modernos se suelen estructurar en cuatro partes muy bien definidas, el *BIOS,* que contiene todas las rutinas y datos internos necesarios para el funcionamiento del siste-

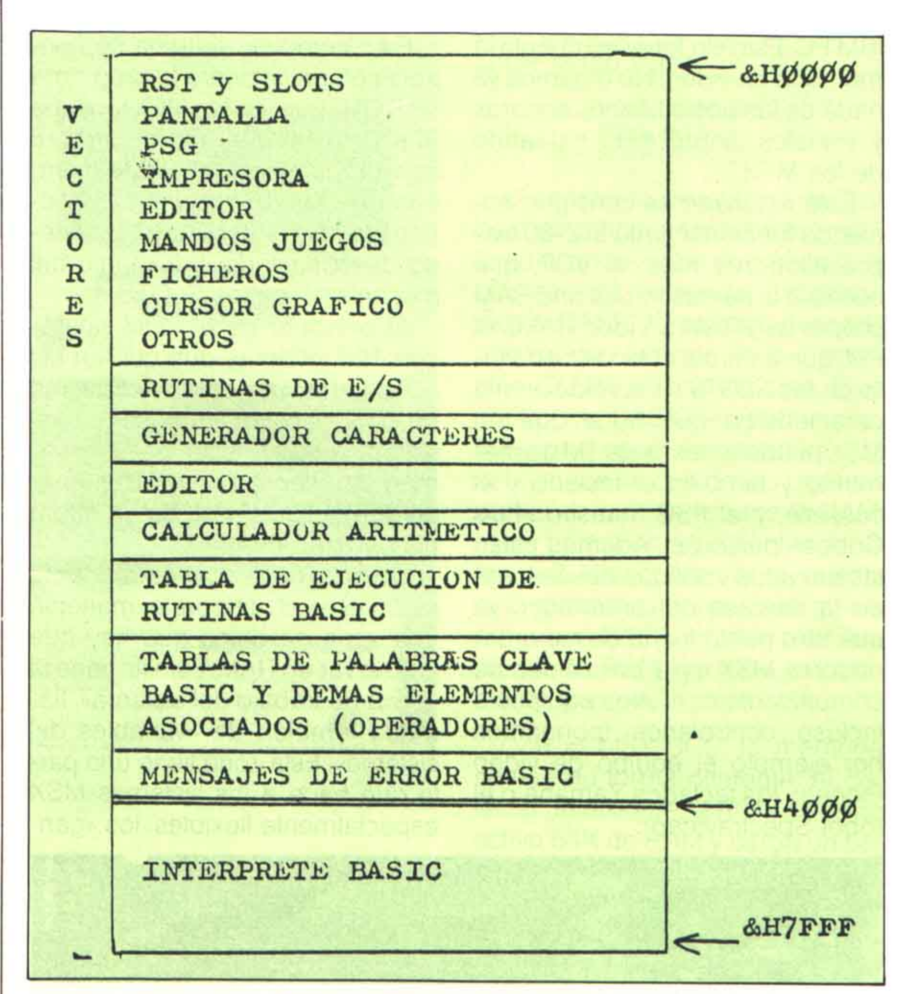


Figura 1: Mapa de la ROM de los MSX1.

ma, el Procesador de Comandos, el Area del Usuario, donde trabajan los programas de éste y se almacenan sus datos, y la Zona Base, por medio de la cual los programas del usuario se comunican con el Sistema Operativo. Esta Zona Base es la clave del funcionamiento de todos los programas que se acogen a un Sistema determinado, ya que siempre está en el mismo sitio y contiene una información perfectamente conocida. Entre otras cosas, aquí está el punto de entrada al Sistema Operativo.

Al poner en un punto fijo la entrada al Sistema Operativo, es posible hacer todo tipo de cambios y adaptarlo a máquinas diferentes sin que los programas que lo usan tengan ningún problema de funcionamiento.

Escribiendo programas bajo un Sistema Operativo

Cuando se escribe un programa en un lenguaje de alto nivel (BASIC, PASCAL...), al programador le importan muy poco los procesos internos de su máquina. El intérprete o el compilador se encargarán de traducirlo al lenguaje interno de la máquina. Una de las grandes ventajas de estos lenguajes es no usar referencias directas a memoria, es decir, cuando tecleamos *PRINT* nos trae absolutamente sin cuidado dónde queda almacenada la instrucción y qué espacio de memoria ocupa.

Al usar código máquina prescindimos del intérprete, con lo que nos vemos libres de sus limitaciones pero también obligados a asumir sus misiones, por lo que tendremos que decidir dónde va el programa y cómo vamos a administrar la memoria. Por oposición a los lenguajes de alto nivel, el código máquina necesita forzosamente referencias directas a memoria; cuando se ejecute una instrucción como JP &H5000, en esa dirección deberá existir un código ejecutable; de lo contrario, el ordenador se bloqueará sin previo aviso (¡aquí no hay *"syntax error"* o cosas parecidas!).

Si JP &H5000 es una llamada a una rutina del Sistema Operativo que envía un carácter a la impresora, esa rutina deberá estar SIEMPRE ahí para que el programa que la usa pueda funcionar. Este problema puede evitarse si el Sistema Operativo cuenta con unas referencias fijas para acceder a las distintas rutinas que lo forman. Hay, en principio, dos formas de usar estas referencias fijas:

1.º— Vectores. Un vector es una instrucción de salto colocada en un punto fijo de la memoria y que contiene la dirección correcta de una rutina. Podemos compararlos a las placas que suele haber a la entrada de los edificios de oficinas, que nos dicen dónde localizar cada una de ellas.

2.º— Funciones. En las llamadas a función hay un único vector que

llama a una rutina a la que se suministra el número que tiene la rutina que queremos usar (este número lo da el fabricante del Sistema Operativo). Esta rutina explora una tabla que contiene las direcciones (punteros) de las distintas rutinas del Sistema, carga la dirección correcta y envía a ella el programa. El trabajo del vector es similar al de un conserje de hotel; su armario de llaves hace las veces de la tabla de punteros.

Según vimos anteriormente, todo programa que use estas referencias fijas podrá seguir funcionando cuando el Sistema Operativo sufra una modificación. Al usarlas, el programador tiene la ventaja de no tener que usar las instrucciones *IN* y *OUT* en ninguna parte de su trabajo. Estas instrucciones tienen el grave inconveniente de depender TOTALMENTE del *hardware*. Esto quiere decir que cualquier cambio en la estructura física del aparato (como ha ocurrido con el nuevo procesador de vídeo de los MSX2) hará que los programas que las usen dejen de funcionar. Si el programa hubiese hecho uso del Sistema Operativo, las modificaciones hechas en este habría hecho que continuase funcionando sin ninguna dificultad.

El sistema MSX

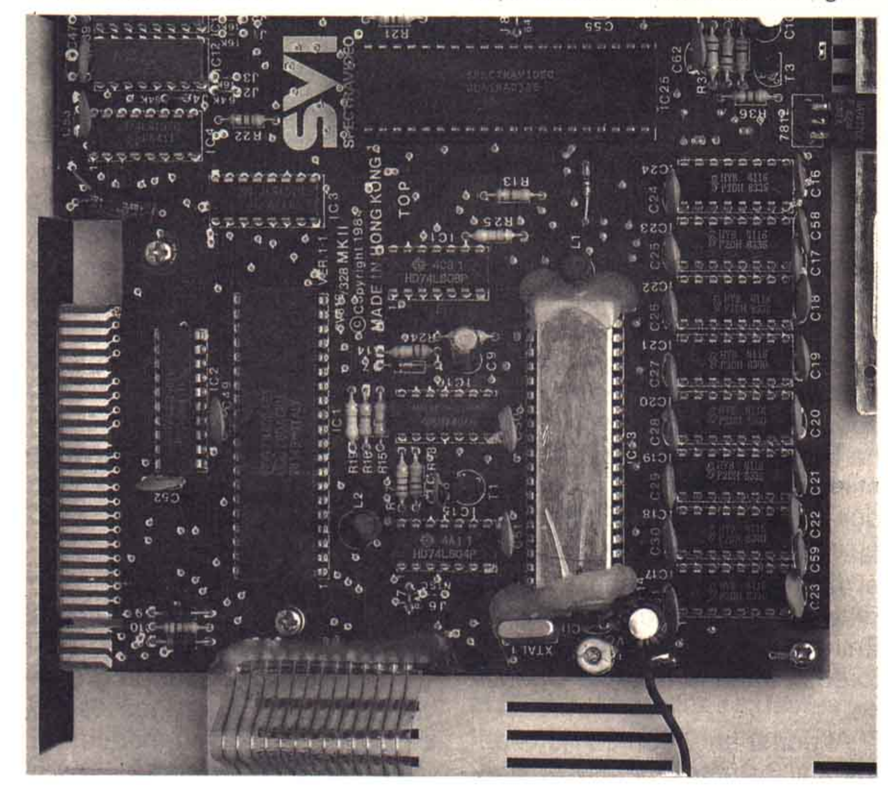
Cualquiera que se haya detenido un poco a estudiar las características generales de un ordenador MSX habrá podido comprobar el gran trabajo hecho por los diseñadores. Con un procesador como el Z-80, que sólo puede direccionar 64K de memoria, se consigue controlar hasta 1 *Megabyte* (¡1.048.576 *bytes*!), que es tanto como la máxima capacidad de un IBM PC. Esto sin tener en cuenta la memoria de vídeo. No digamos ya nada de las posibilidades sonoras y visuales, sobre todo hablando de los MSX2.

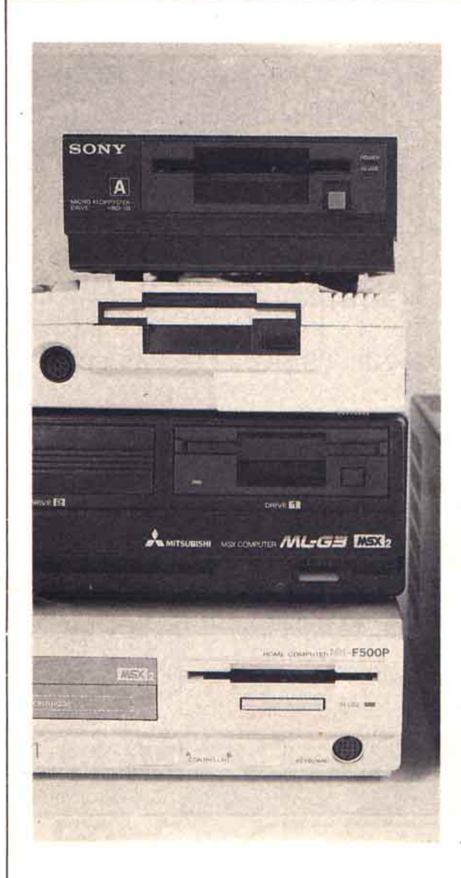
Este resultado se consigue haciendo funcionar junto al Z-80 tres procesadores más: el VDP, que controla la pantalla y usa una RAM propia (la VRAM o Vídeo RAM), el PPI, que controla el funcionamiento de los *SLOTS* (la revolucionaria característica que hace que un MSX pueda tener hasta 1M de memoria) y también el teclado y el cassette, y el PSG, nuestro «Luis Cobos» particular. Además están las entradas y salidas desde fuera de la carcasa del ordenador, ya que otro punto fuerte de los ordenadores MSX es la posibilidad de comunicarse con otros equipos e incluso controlarlos (pongamos por ejemplo el equipo de vídeo Pioneer, los teclados Yamaha o el robot Spectravideo).

Este complejo sistema se controla con un *firmware* (o programa en ROM) que, en los MSX1, ocupa 32K. De estos 32K, los primeros 16 son propiamente el *BIOS*, mientras que los 16 restantes contienen las rutinas del BASIC, pudiendo desconectarse por medio del mecanismo de los SLOTS.

Al principio de la ROM aparecen 108 vectores que cubren las actividades básicas del Sistema (*SLOTS*, pantalla, impresora, cassette, cursor...); ésta es la parte más importante para el programador. En la figura 1 se da un mapa aproximado de la ROM.

Como cualquier otro programa, el *BIOS* y el intérprete manejan una serie de datos que hay que conservar en RAM. Este fin tiene la «zona de trabajo del sistema», llamada también de «variables del sistema». Esta zona tiene una parte que hace a los sistemas MSX especialmente flexibles: los «gan-





chos» *(HOOKS)*. Un gancho está formado, normalmente, por 5 instrucciones *«RET»* del Z-80 seguidas. Varias rutinas de la ROM (especialmente las que se refieren al acceso a dispositivos de almacenamiento externo, como el *diskette*) llaman por medio de una instrucción *CALL* a la primera dirección de cada gancho. Si modificamos su contenido (por ejemplo, con una instrucción JP...) podemos modificar o ampliar la tarea que realiza la rutina. Esto tiene múltiples aplicaciones; por poner un ejemplo, vamos a enganchar la rutina *MAIN*, que es llamada cada vez que vemos aparecer el "OK" en pantalla, para hacer que el ordenador de un *«BEEP»* cada vez que termine un programa o tecleemos una instrucción en modo directo. Esto lo hacemos introduciendo la instrucción *CALL BEEP* en el gancho *H-MAIN*, y puede hacerse como sigue:

```
POKE 65293,192
POKE 65294,0
POKE 65292,205
```

El contenido de la Zona de Trabajo es crítico y no debe manipularse si no se sabe con certeza lo que se está haciendo. En principio, el BASIC la protege evitando que un uso erróneo de la instrucción *CLEAR* pueda alterarla. Bajo control del BASIC, esta zona contiene una serie de variables que definen los límites de las distintas partes en que se reparte la RAM; estas partes se representan en la figura 2. Hay que tener en cuenta que, aunque lo normal sea que el programa BASIC se sitúe al principio de la RAM no tiene forzosamente que ser siempre así.

Con el MSX-DOS, la memoria se usa de forma diferente. Al detectar un *diskette*, la ROM de disco activa 64K de RAM y carga un pequeño programa, llamado *BOOT*, que ocupa el primer sector de todo *diskette* formateado en MSX-DOS. Este programa carga el Sistema Operativo (MSXDOS.SYS) que, a su vez, se encarga de cargar el procesador de comandos (COMMAND.COM); con esta última operación queda activado el Sistema Operativo y aparece el cursor.

La figura 3 representa el mapa de memoria bajo control del MSX-DOS. La Página Cero contiene, en la dirección &H0005, el vector de entrada al Sistema Operativo. La dirección de destino de este vector se llama *BIOS (Basic Disk Operating System)*, y marca el límite superior de la *TPA (Transient Program Area)*, que es la zona de que dispone el usuario para almacenar sus propios programas y los datos que éstos generan. En la Versión 1.0 del DOS, esta dirección es &HD106, lo que da 53254 *bytes* libres para el usuario (la *TPA* comienza en &H0100, que es el límite superior de la Página Cero).

Los programas bajo MSX-DOS que usan la raíz .COM tienen la característica de comenzar siempre en la dirección &H0100.

Problemas con las unidades de disco

Al hablar del MSX-DOS aparece otra de las cuestiones planteadas al enjuiciar la compatibilidad MSX. Esta se plantea de dos formas: la variedad de formatos de los *diskettes* y los problemas que aparecen al conectar el *drive*.

El primero de ellos se discutió en el número 18 de esta revista. El segundo requiere algo más de atención, ya que se plantea continuamente a los usuarios de discos.

Una gran parte de los programas MSX existentes han sido pensados para usarse sólo con cassette; si además añadimos que muchos de ellos han sido hechos para funcionar con ordenadores de 16K RAM, en que ésta ocupa la última página del mapa de memoria (una página tiene 16K), nos encontramos con que la mayoría, y sobre todo estos últimos, invaden la zona de trabajo del sistema de disco. Resultado: es muy normal que un programa en cassette que funciona perfectamente con el disco desconectado deje de hacerlo al conectar el disco. También es frecuente que el ordenador quede bloqueado al intentar cargar estos programas.

Los comentarios que suelen darse en estos casos son: «el disco no es compatible» o «¿para

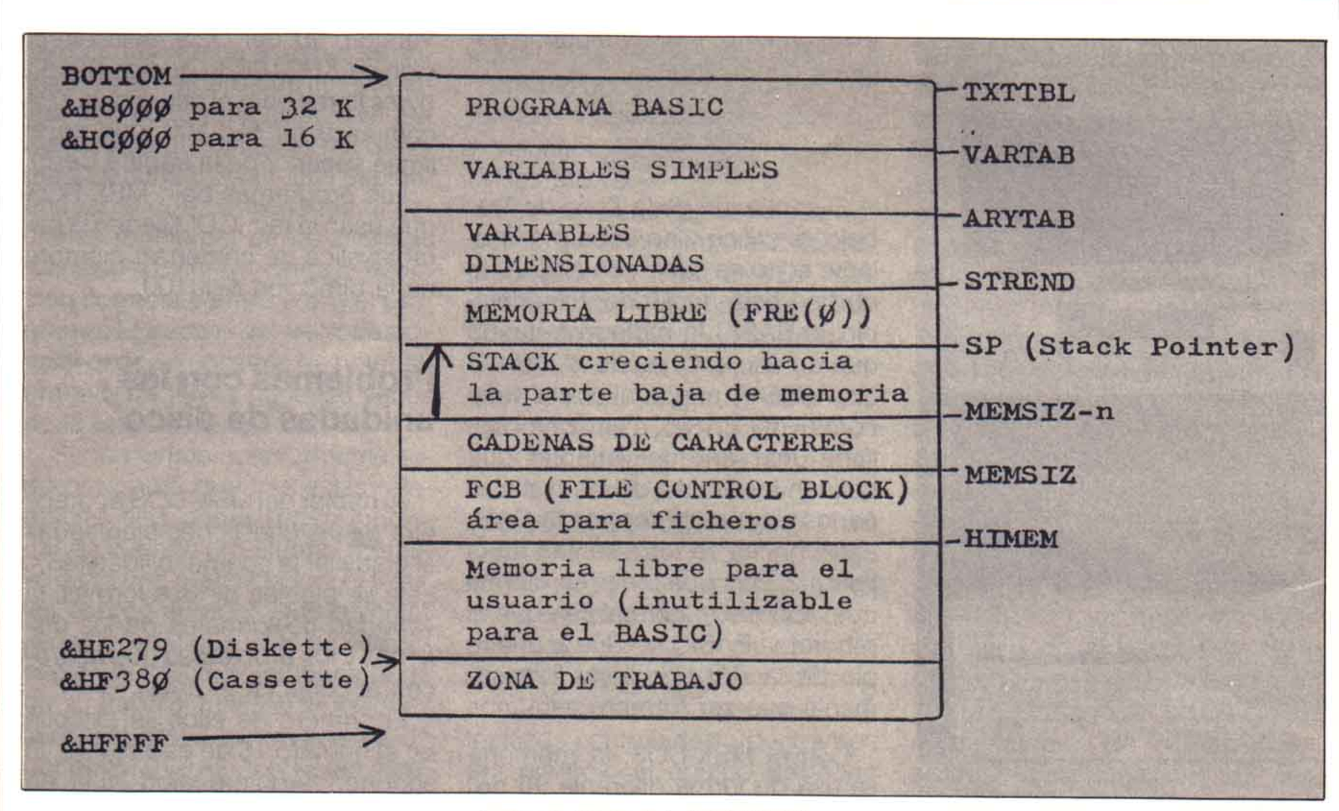


Figura 2: Mapa de la RAM BASIC.

qué sirve el disco si no puedo aprovechar su rapidez grabando en él mis programas en cassette?». La respuesta es, nuevamente, «el fallo está en los programas, no en el ordenador».

Algunos usuarios pueden solucionar el problema simplemente desconectando la unidad de disco. Otros no lo tienen tan fácil, como ocurre con los que tienen un Spectravideo X'PRESS. Suele recurrirse a teclear la instrucción CLEAR 100,&HF380, pero esta solución sólo es válida si el problema se debía a que el programa se «tragó» el *STACK* mientras cargaba. Ahora bien, el problema puede haber sido causado por la destrucción de una rutina llamada por medio de un gancho (todo el sistema de disco funciona con ganchos). La mejor solución es anular totalmente el disco con el siguiente programa:

```
10 CLEAR 100,&HF380
20 FOR N=64000 TO 64015
30 READ A$:POKE N,VAL("&H"+A$)
40 NEXT N
50 DEFUSR=64000:A=USR(0)
70 DATA F3,01,2F,02,11,9B,FD,21,9A,FD,
   36,C9,ED,B0,FB,C9
```

En ensamblador sería:

```
FA00H DI            ; inhibe
                      interrupciones
      LD BC,022F    ; longitud tabla=559
                      bytes
      LD DE,FD9B
      LD HL,FD9A    ; inicio tabla
      LD (HL),C9    ; coloca un "RET"
      LDIR
      EI            ; reactiva
                      interrupciones
      RET           ; vuelve al BASIC
```

Este programa puede ser grabado en disco. Antes de teclear *RUN* debemos comprobar que la luz que indica el funcionamiento del drive está apagada, si no quedaría girando continuamente el disco, lo que puede dar lugar a una avería.

En cuanto a pasar los programas en *cassette* a *diskette*, el procedimiento es más complicado. En líneas generales, los pasos a seguir son los siguientes:

1.º Cargar y grabar los distintos bloques de código máquina.

2.º Cuando uno de estos bloques pueda invadir la zona de trabajo del disco, cambiar su dirección con la instrucción

BLOAD"CAS:nombre",HEX$(dirección inicial-dirección de carga)

donde la dirección «inicial» es la que realmente tendría si se cargase normal normalmente.

3.º Preparar y grabar un programa que cargue los distintos bloques, permita parar el disco, lo anule, coloque los distintos bloques en los lugares adecuados y arranque el programa principal.

Cómo hacer programas transportables para MSX

Con lo visto hasta ahora tendremos ya una idea general de cómo un programa puede fallar al cambiar de máquina y otro puede funcionar sin dificultad, al haber sido planeado para ser TRANSPORTABLE. Vamos a concretar esto un poco centrándonos sobre los sistemas MSX.

En BASIC, un programa que no use referencias directas a memoria (*CLEAR,POKE,* llamadas a código máquina....) debe trabajar perfectamente en cualquier ordenador MSX que tenga una capacidad de memoria adecuada a su longitud. Si se van a usar estas referencias se deben tener en cuenta los principios que se van a dar para los programas en código máquina. Si se usa la instrucción *CLEAR* debe tenerse en cuenta que, si se usa un valor superior a &HE279, peligra el funcionamiento del programa al usar *diskettes.*

Al programar en código máquina debe hacerse lo siguiente:

1.° Usar los vectores para toda operación que represente entrada o salida. El usar *IN* o *OUT* (*INP* y *OUT* en BASIC) hará que el programa funcione con plena garantía solamente en la máquina en que fue preparado.

2.° En lo posible, respetar las zonas de trabajo del sistema. Con una memoria de 64K hay espacio suficiente para enormes sofisticaciones. Si, además, añadimos que es posible usar al mismo tiempo el *BIOS* y la RAM que hay «detrás» de él (gracias a los vectores de control de los *SLOTS*) podemos decir que tenemos 16K adicionales.

Para programas a usar con MSX-DOS, las reglas son iguales. Todas las operaciones de entrada/salida deben hacerse a través de lamadas al vector *BDOS.* En cuanto a la extensión del programa y su zona de trabajo, deben estar dentro de la *TPA.* Si hace falta puede usarse el espacio que ocupa el Procesador de Coman-

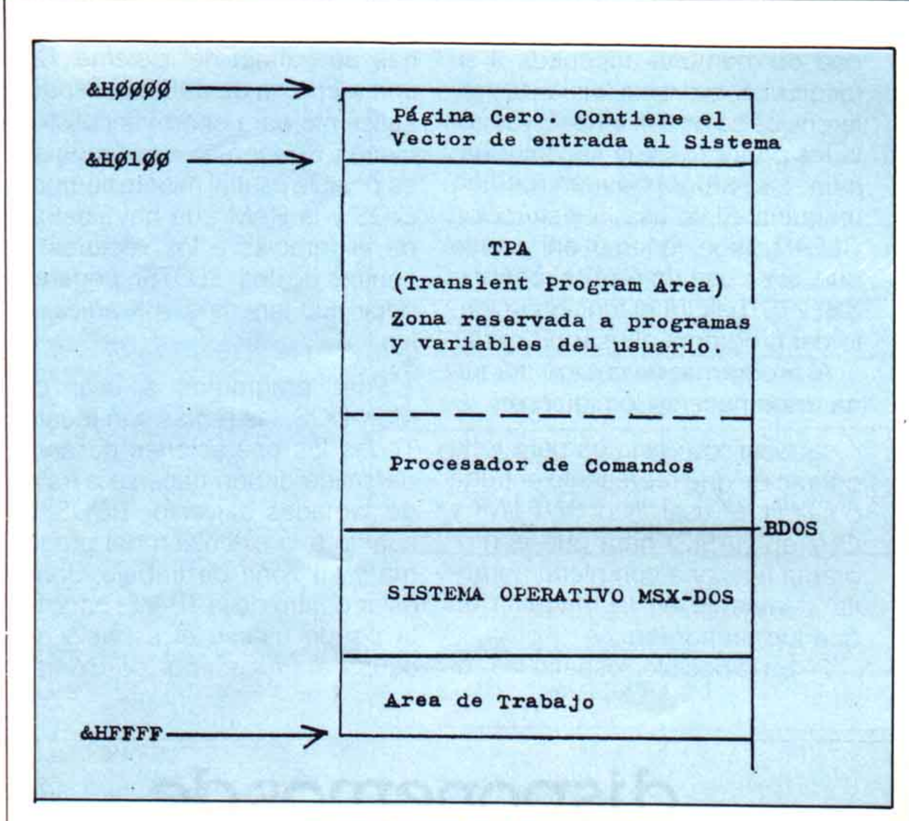


Figura 3: Mapa de la RAM MSX-DOS.

dos. Al terminar el programa haremos una llamada al vector *WBOOT* (&H0000) para volverlo a poner en memoria.

Casos particulares

No siempre es posible hacer un programa que respete todos los principios expuestos; éste es, precisamente, el caso de los auxiliares. Al ser su misión principal controlar las operaciones de entrada y salida de cada máquina en particular, dependen en todo de las características de ésta. Por ello es perfectamente normal, y no indica en absoluto falta de compatibilidad, que el Sistema Operativo de una máquina no funcione en otra. Esto ocurre con el CP/M del Spectravideo X'PRESS. Este ordenador puede trabajar en 80 columnas; al estar el CP/M adaptado a esta modalidad no tiene por qué funcionar en máquinas que usan sólo 40 columnas.

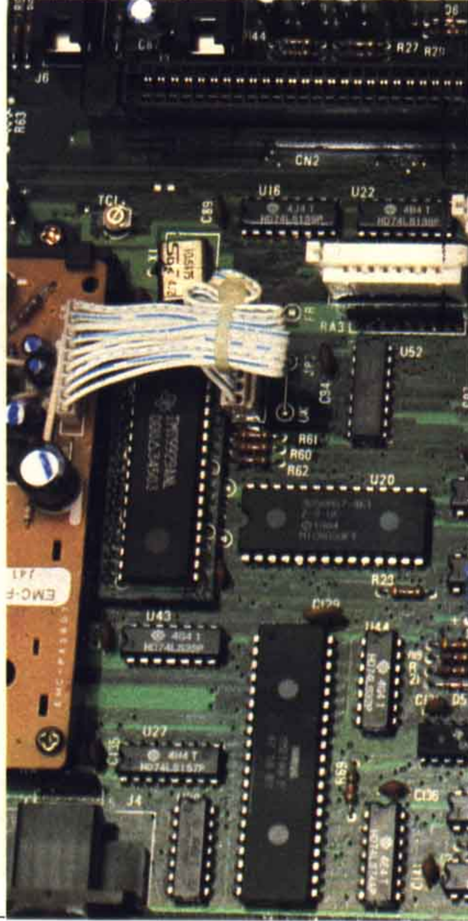
También debe considerarse que, en un campo que evoluciona tan rápidamente como el de la Informática, todo sistema que no pueda ser mejorado está condenado al fracaso. Por ello se ha introducido el concepto de «compatibilidad ascendente», que supone que las nuevas versiones de un Sistema Operativo permitirán que cualquier programa preparado con una versión anterior pueda funcionar con ellas; sin embargo, el supuesto contrario no es válido. Concretando: todo programa para un MSX1 debe funcionar en un MSX2, pero no al revés, lo que no es extraño si tenemos en cuenta las nuevas capacidades de los MSX2.

Incompatibilidades en la práctica

Como término a estas observaciones vamos a estudiar un par de casos particulares de incompatibilidad que, después de un pequeño estudio, demuestran que el fallo está en el poco conocimiento, por parte del programador, de las características del Sistema MSX.

El primero se refiere a un programa que, independientemente del error que veremos, es excepcionalmente bueno: el ajedrez CHESS'86 de PHILIPS. Este programa ocupa 32K de RAM y hace uso del *BIOS* para sus operaciones de entrada/salida.

Este programa cuenta con una



presentación y una calidad de juego que le hacen estar entre los mejores para cualquier tipo de ordenador. Sin embargo, tiene un fallo que habrá notado cualquiera que haya tratado de cargarlo en un Spectravideo X'PRESS: el ordenador queda bloqueado a poco de iniciarse la carga y hay que pararlo para recuperar el control. Sin embargo, con una ligera modificación en el programa cargador, el programa funcionará a la perfección. ¿Cuál era el fallo? Lo primero que hace el cargador es modificar el mapa de memoria, inhabilitando el BASIC y sustituyéndolo por RAM; ahora bien, esa RAM ocupa *SLOTS* diferentes en ambos ordenadores, cosa no prevista por el programador.

Veamos ahora un programa de la firma «Software Projects». Los programas de esta casa suelen caracterizarse por sus sofisticadas protecciones, de las cuales la más moderna es el típico código de colores. Otras protecciones menos visibles consisten en quitar la cabecera de los programas o usar sistemas de carga propios. Veamos el caso particular de *«Jet Set Willy II»*. Este programa comienza con un cargador en BASIC y continúa con otro cargador en código máquina. En el cargador BASIC está el primer fallo. Al usar la instrucción BLOAD"M" para cargar el siguiente bloque, en vez de BLOAD"CAS:M", la carga no puede hacerse si hay unidad de disco conectada, ya que el Sistema intentará buscar «M» en el *diskette*. Una vez corregido esto, el problema queda subsanado. Sin embargo, estudianto el cargador de código máquina encontraremos algunas cosas que hacen dudar seriamente que el programa pueda funcionar en cualquier MSX1 y, por supuesto, que lo haga en un MSX2. Este cargador cambia el modo de interrupciones del Z-80 de 1 (que es el normal) a 2. Usa, además, un sistema propio de acceso a pantalla a través de instrucciones *IN* y *OUT*. Por último, la rutina de carga desde *cassette* es también propia.

Todo programa MSX1 que no use vectores *BIOS* para acceder a pantalla es más que improbable que funcione en un ordenador MSX2. Un solo hecho basta para mantener esta afirmación. Un MSX1 tiene una VRAM de 16K, por lo que las direcciones deben darse como números de 2 bytes, mientras que un MSX2 tiene 128, con lo que hacen falta 3 bytes para dar esas direcciones. El *BIOS* de un MSX2 puede tener prevista la conversión; pero, si las direcciones se pasan a *BASE* de *IN* y *OUT*, el fallo es inevitable.

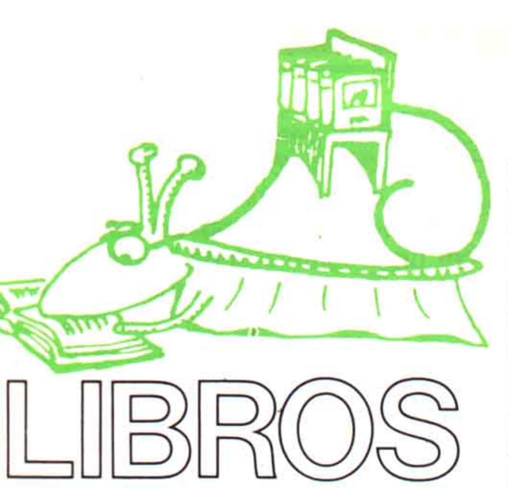
A modo de reflexión final, al hacer un programa debe huirse de técnicas extrañas a su propia finalidad. Algunos «descubrimientos espectaculares» pueden hacer peligrar la calidad del programa si no están suficientemente comprobados. Por otra parte, derrochar esfuerzo e imaginación en sofisticados sistemas de protección no solo significa una preciosa pérdida de tiempo sino también, a veces, un perjuicio para el resultado final. Volviendo al *«Jet Set Willy»*, nada hay más fastidioso que, después de sufrir la lentitud con que carga el programa y fallar dos veces al introducir el código de colores, quede borrado de la memoria el juego.

Es un hecho archidemostrado que, no importa cuál sea el tamaño del ordenador, toda protección por *software* está condenada a la aparición de otro programa que la anule. Lo normal es que, si el programa es bueno, haya múltiples personas buscando la forma de desprotegerlo. Si el programa es «corrientito», lo más seguro es que los usuarios no se molesten en investigar y, simplemente, lo desechen.

Como muestra de la poca utilidad de estos sistemas, recomendamos a aquellos propietarios del *«Jet Set Willy II»* que estén cansados de sufrir el «suplicio de los colores» cargar el programa de la siguiente forma:

```
10 CLEAR 100,34303
20 BLOAD "CAS:"
30 POKE 35937,135
40 DEFUSR=36155:A=USR(0)
RUN
```

Lorenzo Hernández Talavera



LIBROS

Libro: Estadística: Teoría, problemas y aplicaciones en BASIC
Autores: J. Tennant - Smith
Editorial: Anaya
Páginas: 218

La importancia de los ordenadores y de las técnicas de cálculo difícilmente puede ser sobrevalorada en nuestros días. El poder de los ordenadores en la ciencia, tecnología, industria y comercio es universalmente reconocido, pero quedan aún enormes potenciales por desvelar, a consecuencia de la falta de conocimientos sobre probabilidad y estadística.

Los especialistas en estadística han realizado importantes avances mediante el uso y aplicación de los ordenadores a las técnicas de análisis multivariante y el análisis de las series cronológicas. Estas series habitualmente se desarrollan en ordenadores con lenguajes de alto nivel, como el FORTRAN, PASCAL u otros lenguajes matemáticos especializados. Queda, no obstante, un amplio campo para los programas de probabilidad y estadística en lenguaje BASIC, aplicados a problemas que son matemáticamente más sencillos pero de una indudable importancia práctica.

En este libro los autores se han propuesto desarrollar una labor didáctica muy seria con un doble objetivo. Por un lado, introducir a los neófitos en la programación en el lenguaje BASIC, y por otro, presentar una enseñanza gradual y ordenada sobre la estadística, comenzando por la estadística descriptiva elemental, pasando por la teoría elemental de la probabilidad, experimentos de simulación, variables aleatorias, distribución de probabilidades, funciones de densidad, muestreo y estimación, etc., y llegando a la regresión y correlación, para culminar la obra



con otras técnicas y modelos estadísticos.

Todo este recorrido por la ciencia de la estadística está muy bien acompañado por ejemplos de programas BASIC aplicados a cada concepto desarrollado, así como un gran número de propuestas para variar los ejemplos dados y para resolver problemas mediante programas sencillos en el ordenador.

Aunque los listados que aparecen en el dibujo están desarrollados en una versión de BASIC Microsoft, el primer capítulo está dedicado íntegramante a explicar las peculiaridades de otros dialectos BASIC y el camino a seguir para que cada lector realice la adaptación de los ejemplos propuestos al BASIC que incorpore su microordenador.

Ahora que, con el proyecto Atenea, parece que los ordenadores van a estar cada día más presentes en las escuelas, libros como éste, excelente en su planteamiento y su desarrollo, irán cobrando cada vez mayor importancia, ya que pueden resultar excelentes ayudantes, tanto para el profesor con conocimientos de estadística pero neófito en la programación BASIC, como para el estudiante que ya conozca dicho lenguaje y esté presto a aprender estadística.

Libro: Introducción a MSX LOGO
Autor: Manuel Medel
Editorial: Peopleware, S.A.
Páginas: 124

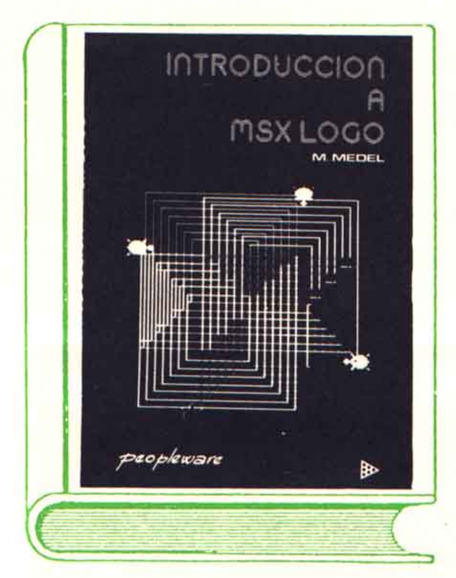
La denominación que puede definir al LOGO es adaptabilidad, ya que, a diferencia de lo que ocurre con otros lenguajes como el BASIC —donde los comandos e instrucciones forman parte de su intérprete y no pueden variarse—, el LOGO, por tratarse de un lenguaje procedimental, nos permite crear procedimientos adaptados a nuestras necesidades y que podemos usar como si fueran instrucciones. Con estos procedimientos podremos realizar un lenguaje de programación adaptados a los problemas que quera-

mos resolver, y lo que es más importante, a nuestra forma de pensar.

También es muy conocida la capacidad de los gráficos de tortuga del LOGO, que lo hacen muy popular entre los jóvenes, así como su capacidad interactiva y estructural que lo hacen ideal para iniciar a los niños en la informática.

Es precisamente en esta dirección en la que está orientado este libro, excelente para aprender LOGO, y que además presenta la ventaja de estar basado en el LOGO PHILIPS MSX, con lo cual los ejemplos pueden teclearse directamente sin necesidad de adaptarlos de un dialecto a otro.

Tras una breve introducción en la que se explican los conceptos generales necesarios (procedimientos, primitiva, etc.), comenzamos a aprender los nombres y funciones de las diversas primitivas, agrupadas por capítulos según sus características comunes (las de movimiento de tortuga, las de manejo de sprites, las de manejo de cinta/disco, etc.).



En todos los casos se plantean ejemplos, primero sencillos y progresivamente más complejos, que nos ayudan a familiarizarnos con el lenguaje y a memorizar la función que desarrolla cada primitiva.

De este modo, podemos aprender a manejar esta excelente versión de LOGO (muy rápida, muy completa y en castellano) con nuestro MSX. Un libro muy interesante.

SOFTWARE

Programa: The Goonies
Tipo: Juego
Distribuidor: SERMA
Formato: Cartucho

Basado en un fantástico *film* cinematográfico, SERMA nos presenta este juego, *The Goonies,* una maravillosa aventura con una divertida trama y un apasionante final.



El juego para aquellos que no conocéis la película, os será aún más entretenido. Nos encontramos en un pueblecito costero, *Cauldron Point,* en un paraje desconocido llamado los muelles de *Goon,* cuyo entorno encierra un fascinante secreto.

The Goonies son un grupo de amigos en busca siempre de aventuras, ese afán por buscar experiencias nuevas les ha llevado a encontrar un tesoro, reflejado en un mapa por el pirata más temido de la región Willy «el tuerto». Dispuestos a ir en su busca llegarán a la guarida de los hermanos Fratelli, los cuales desconocen la existencia de este tesoro pero ocultan cosas más peligrosas, juego ilícito, etc.

El juego comienza en el momento que os encontráis a uno de los hermanos Fratelli, Sloth, por quien seréis representados a lo largo del juego, ya que él no es como sus hermanos y ha decidido unirse al grupo. Vuestras vidas comienzan a correr peligro. Sloth, se encontrará con un sin fin de cavernas y mazmorras, y lo que es peor, a sus hermanos tras él y monstruos que habitan por los pasadizos y que fueron los compañeros del pirata Willy.

Este juego está preparado para un solo jugador y la acción puede controlarse tanto con *joystick,* como con los cursores.

Una interesante peculiaridad de este juego es que debes tener presente el nivel de vitalidad y experiencia, que se verá reflejado en dos barras en la parte superior de la pantalla, las cuales irán creciendo y/o decreciendo a medida que avanzamos por las cavernas y masa a los monstruos. Elemento principal en el que debe fijar su atención, son las llaves que irán apareciendo a su paso, las cuales debe recoger, eso sí, recordando que sólo podrás transportar una cada vez. Estas llaves te permitirán abrir las mazmorras que encuentres a tu paso y así salvar a los *Goonies,* pues han sido encerrados. Cada vez que logres salvar a los siete *Goonies* acabará una etapa, pero sólo podrás acceder a la siguiente si has conseguido todos los elementos necesarios y se encuentres en poder de la contraseña.

Elementos necesarios son: zapatillas, eso sí, especiales, ya que con ellas podrás correr más deprisa. El casco; los lugares que visitarás, sin duda serán muy abruptos, por eso es necesario que lo encuentres y comida que te dará la vitalidad suficiente para seguir adelante.

El juego consta de cinco etapas subdivididas en cinco escenas a su vez, están interconectadas por unas entradas calavéricas, así podrás investigar todas y salvarte más de una vez.

Para todos aquellos seguidores de Konami, no hace falta que digamos nada más del juego, ya que su calidad está demostrada desde el primero al último, pero sí nos gustaría comentar la innovación que supone en el mundo del *software* llevar un *film* cinematográfico al ordenador, ya no por las técnicas empleadas, que sin duda son excelentes, sino por ese afán siempre de motivar a los usuarios de ordenador, lo cual agradecemos.

Destacar la música que aparece a lo largo del juego y la sensibilidad de movimientos

Puntuación:
Presentación: 8
Claridad: 8
Rapidez: 9
Adicción: 9

Programa: Livinsgtone, Supongo
Tipo: Juego
Distribuidor: Opera Soft
Formato: Disco de 3.5"

¡¡Fantástico, sensacional!! Opera Soft ha dado en el blanco de las tendencias videoadictas. Livingston Supongo, está llamado a convertirse en un mito en el mundo de los juegos de ordenador.

La versión que hemos obtenido para los ordenadores MSX, no tiene que envidiar nada a las anteriores creadas para otros ordenadores. El fabuloso y singular tema del que se trata y el cual forma la aventura que ahora os voy a narrar es de las más interesantes, tanto en la realidad como en la ficción, la cual ahora llevada al mundo de los juegos de ordenador hará las delicias de más de un usuario.

Permitid que me presente, soy Morton Stanley. Me establecí en Nueva York, y fue allí donde empezó todo, la más increible historia que un hombre jamás pudo contar o imaginar.

Me enviaron como corresponsal a Africa en busca del paradero del Dr. Livinsgton, la selva era inimaginable y cada vez que me adentraba más en sus caminos todo se volvía más misteriosa y espeluznante.

Como podréis suponer, vuestra misión es la de encontrar al Dr. Livinsgton y referir unas palabras si aún os queda aliento para realizar esa labor, cuando hayáis conseguido pasar todas las pruebas a las que os veréis sometidos.

No sólo necesitaba víveres, sino que también armas con las que pudiera defenderme de los numerosos peligros con los que sin duda me encontraría. Me hice como pude un fuerte Boomerang, unas lanzas y algunos cócteles Molotov, con el whisky que aún me quedaba, además de una pértiga que me cedió uno de los lugareños con cara de pocos amigos.

Como podréis figuraros, sólo contaréis con estas armas a las que he hecho referencia anteriormente, la aventura que estáis viviendo está llena de inverosímiles hazañas las cuales efectuaréis, ya que el juego está formado por numerosas pantallas que se suceden y las cuales os van acercando a Livinsgton. Pero a lo largo del camino os iréis encontrando cinco piedras preciosas guardadas fielmente por las tribus de la zona o escondidas en lugares que jamás habíais imaginado, debéis recogerlas y transportarlas hasta la penúltima etapa del juego, donde las debéis ofrecer dentro del Templo.

Si llegáis a superar esta etapa estaréis más cerca que nunca del fin pero no os confiéis ya que a medida que se desarrolle el juego, veréis que hay un principal enemigo vuestro, un buitre que en el caso de que os atrape os llevará hasta el principio del juego y tendréis que empezar de nuevo.

Cuando llevéis bastante aventura recorrida debéis daros cuenta que vuestra energía se acaba, encontraréis a lo largo del juego botijos y barras de pan que os permitirán seguir a adelante.

Aunque nos esforcemos en describiros este fascinante juego, no existirían palabras para ello, es por eso que nadie mejor que Ud. puede comprobarlo. De la realización gráfica y estructura interna del programa, cualquier elemento sería digno de mencionar. Podemos destacar entre todos principalmente la maravillosa melodía, que escuchamos al pasar la prueba del Templo, generada como si de un instrumento se tratase.

La historia dice, que cuando vi por primera vez a Livinsgtone le dije, como un buen flemático inglés: «Livingstone, supongo.».

Puntuación:
Presentación: 9
Claridad: 8
Rapidez: 9
Adicción: 10

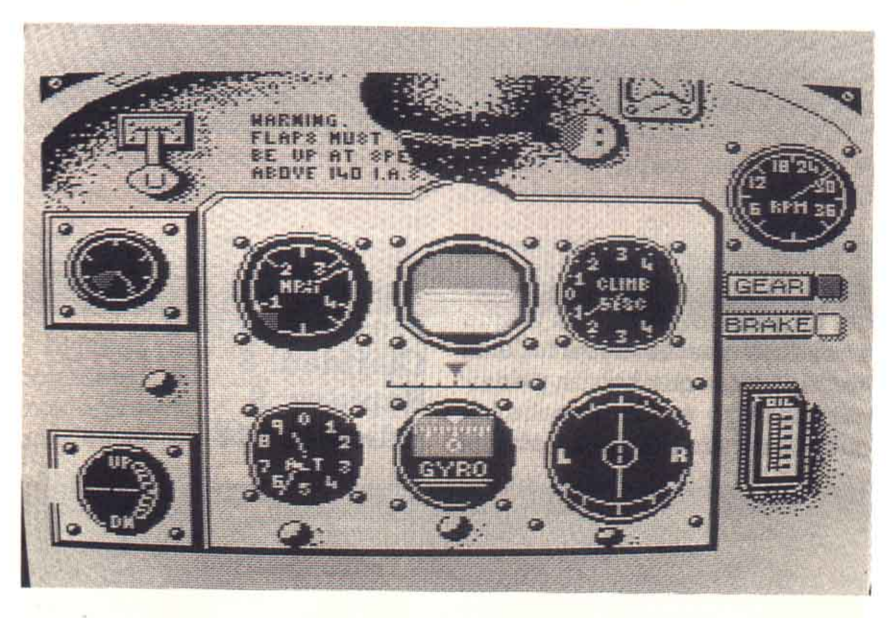


SOFTWARE

Programa: Spitfire 40
Tipo: Juego
Distribuidor: Mirrorsoft
Formato: Cassette

Spitfire 40 no es sólo la mejor aventura que puedas tener volando, sino que este simulador de vuelo, en toda la extensión de su contenido, es la más espectacular acción de guerra.

Nos han llegado aquí directamente de Inglaterra, por lo que hemos podido ver su originaria presentación y desarrollo el cual como simulador tiene todas las características necesarias y más que suficientes.



El juego se presenta con un manual de instrucciones por el momento en inglés, hecho que no dificulta la acción, ya que la disposición de los elementos en la pantalla, nos resulta muy clara a la hora de pilotarlo.

Los combates aéreos siempre han sido un hecho un poco lejano a nosotros, y los programas que hasta ahora había en el mercado no reunían todas nuestras necesidades.

Spitfire 40, es sin duda lo que muchos esperábamos, dejando siempre una puerta abierta a la superación. Un juego en conjunto interesante, la disposición de la cabina de mandos, en la que podemos observar con claridad todos los niveles de velocidad, altura e inclinación del aparato junto a otros muchos elementos imprescindibles para realizar la misión.

Un mapa de vuelo en el que nos indica la posición nuestra, del adversario y de los aeropuertos más cercanos, nos facilitará el vuelo y el objetivo. Y lo más importante el punto de mira de la ametralladora que nos servirá para batir a nuestros enemigos. Comenzar la misión sólo depende del despegue que efectuemos, potencia, velocidad y precisión, serán los elementos necesarios. Pero como no todos nacemos sabiendo, este programa cuenta con tres opciones, una primera denominada prácticas en la que podemos efectuar el despegue y aterrizaje tantas veces como queramos dirigiéndonos al objetivo que más nos guste. Una segunda, propia de combate en la que mediremos nuestra astucia con numerosos enemigos y que nos servirá para familiarizarnos con el aparato y una tercera en la que se desarrolla propiamente el juego y en la que se debe llevar a cabo una misión.

Esta tercera etapa consistirá en alcanzar un objetivo y hacerlo desaparecer, por ello se nos designará un nombre, el cual podremos elegir, el juego, combate o prácticas y los datos del objetivo, posición, altitud, dirección y velocidad.

Habernos familiarizado con el avión es imprescindible ya que cuando estemos en vuelo no podremos quitar ojo de la situación, ver el mapa de situación y mantener el avión es lo más importante. Realizaremos numerosos virajes y el enemigo siempre estará a nuestra cola, desapareced de su campo de acción y no dejéis que se os escape.

En plena batalla todo sucede muy rápido, pero el juego en sí es un tanto lento, sobre todo para aquellos que adaptéis la opción de prácticas, pues tardaréis en llegar a otro aeropuerto. Podréis jugar todo el tiempo que deseéis pues no tenéis nivel de gasolina y el fuego enemigo muy pocas veces os derribará.

Es un verdadero simulacro de vuelo de guerra, numerosos libros han reflejado sus características y se han referido a él, mejor garantía no hay.

Puntuación:
Presentación: 9
Claridad: 8
Rapidez: 6
Adicción: 8

Programa: Samantha Fox
Tipo: Juego
distribuidor: SERMA
Formato: Cassette

El humo ciega mis ojos, pero intento adivinar con la mirada qué juego lleva el contrario, apuro de un trago el *Whisky,* mientras contemplo mis naipes, parecen enviarme un mensaje, quizás mi suerte haya cambiado. Ya no hace falta depender de otras personas para disfrutar de esos juegos de siempre, como es una buena partida de cartas y en este caso, de Póker.

Los distintos juegos de mesa que se nos presentan para la gama de ordenadores MSX, hacen que muchas veces podamos jugar una partida entretenida sin necesidad de llenar la casa de amigotes. *Samantha Fox,* es una de las versiones más entretenidas y a la vez más divertidas, ya que incluye en la misma *cassette* dos versiones del tan temido a veces juego del Póker y a su vez tan enigmático, pues no sólo encierra inteligencia e intuición sino la paradójica situación de la suerte.

Serma, deleita la vista de los varones con un juego de póker muy sexy, en la cara A de *Samantha Fox,* encontramos una versión, en la que la propia *Samantha* será nuestro adversario pero aquí no se jugará con dinero sino que *Samantha* se irá desprendiendo de sus prendas a medida que vaya perdiendo y ya no le quede más ropa, momento en el que habrás ganado y será recompensado con una estupenda foto de póster de Samantha sin un solo atavío. Aunque parezca un poco libertino, la imagen que recogemos en nuestro monitor es digitalizada, es decir, de una imagen de carne y hueso, se ha obtenido esta fotografía, sin duda atractiva. Lo sentimos por el sexo femenino ya que no han tenido la delicadeza de adjuntar otra versión cuyo adversario fuera hombre.

La cara B del juego, se establece como una partida normal en la que existen apuestas y varios contrincantes, siendo en total cuatro jugadores. La modalidad de juego es el póker descubierto de siete cartas, de las cuales a la hora de la verdad sólo se podrán combinar cinco, aunque eso lo hace el programa por ti. Tienes varias opciones, pasar si no llevas juego, ver como reacciona el adversario, cubrir la apuesta para ver las cartas y subirla si la suerte te ha sonreido o vas de farol, y todo esto con dos teclas cursores y *enter.*

En esta segunda parte del juego no podremos ver a nuestros adversarios ya que no se ha incluido ninguna imagen por lo que en comparación con la cara A, pierde un «poco» de interés, tan sólo podremos ver nuestras cartas y las combinaciones a las que podemos optar referidas por la máquina, pues sólo hay un descarte. Aparecen en la pantalla junto con las cartas las apuestas que realizan los demás adversarios.

Hacer referencia a la música del juego, la cual oiremos al principio y al final de éste, y que es la banda original de la película «El Golpe».

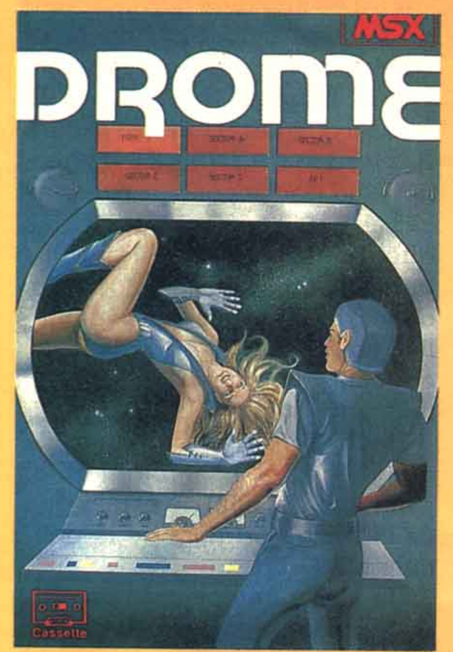
Un juego en conjunto divertido y sorprendente, un poco lento en su desarrollo, pero esto permite mayor concentración.

Suerte... y no desepereís.



Puntuación:
Presentación: 8
Claridad: 7
Rapidez: 6
Adicción: 9

el mejor softwa





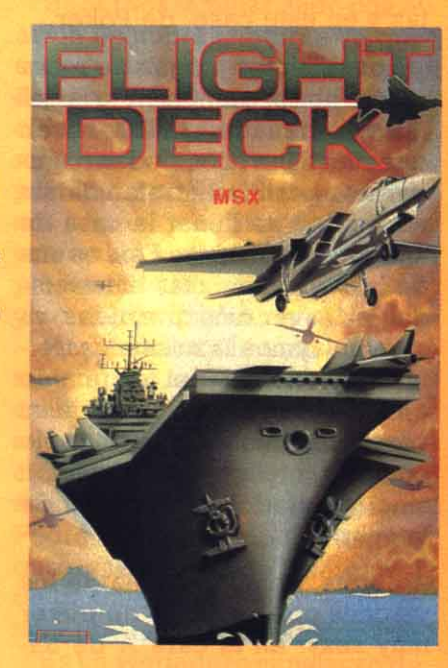
DROME

Entretanto en DROME, un Super-ordenador, debes encontrar y eliminar los sofisticados sistemas de defensa y supervivencia.

Has de elegir uno de los cuatro sectores que constituyen los mecanismos de defensa de esta terrorífica máquina.

Un atractivo juego de acción, donde se pone a prueba la capacidad de la máquina y del jugador.

Precio de venta 2.000 ptas. (**IVA incluido**)



FLIGHT DECK

Sienta la emoción del golfo de Sidra en casa.

FLIGHT DECK es un juego de estrategia y habilidiad en el que tendrás que desmantelar las bases enemigas.

Al mando de un portaaviones donde dispones de 10 unidades de combate... y poco tiempo.

Precio de venta 2.000 ptas. (**IVA incluido**)





MC-ATTACK

Ayuda a Fredy, el Rey de la Hamburguesa a preparar el suculento manjar que hace las delicias de los comensales.

Ten cuidado con las salchichas grasientas y los huevos escurridizos que intentarán arruinar tu exquisito plato.

Defínete con la pimienta y procura hacer el mejor número de hamburguesas posible.

... Buen provecho.

Precio de venta 750 ptas. (**IVA incluido**)

re para MSX



CONFUSED?

Es el puzzle electrónico.

El objeto del juego es resolver 10 puzzles con distinto número de piezas, a elegir, pero todas... MOVIENDOSE.

Pon a prueba tu inteligencia y capacidad de deducción para solucionar algunos de estos entretenidos rompecabezas.

Precio de venta 2.000 ptas. (IVA incluido)

NORTH SEA HELICOPTER

Una explosión en una plataforma en el mar del Norte arroja a los hombres a un destino incierto...

Empieza una carrera contrareloj para salvarles de su fatal situación.

Tienes que convertirte en un piloto experimentado para mantener el control del helicóptero... El tiempo empeora.

¿Crees que cumplirás la misión?

Precio de venta 2.000 ptas. (IVA incluido)









SPACE RESCUE

Desciende de tu nave nodriza a la superficie lunar e intenta rescatar a los hombres extraviados.

Ojo con los meteoritos que te destruirán cuando intentes regresar.

Disponer de nuevas plataformas para culminar con éxito la misión.

Desgraciadamente tu nave nodriza está bloqueada por unidades de combate enemigas... Intenta destruirlas.

Precio de venta 750 ptas. (IVA incluido)

Solicita tu programa favorito en tu tienda especializada habitual o recorta y envía este cupón a INFODIS, S. A., c/Bravo Murillo, 377. 5.º A - 28020 MADRID.

NOMBRE ______

DIRECCION ______

POBLACION ______ PROVINCIA ______ C. P. ______

Forma de pago: CHEQUE ☐ CONTRAREEMBOLSO ☐

TARJETA DE CREDITO: American Express ☐ VISA ☐ INTERBANK ☐

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

CADUCIDAD ______

Firma ______

TITULO	CANTIDAD	TITULO	CANTIDAD
DROME FLIGH DECK MC-ATTACK		CONFUSED? NORTH SEA HELICOPTER SPACE RESCUE	

infodis, s.a.

SOFTWARE

Programa: Cosa Nostra
Tipo: Juego
Distribuidor: Opera Soft
Formato: Disco de 3,5"

El superdetective *Mike Bronco* de gran prestigio mundial ha sido contratado por el alcalde de Chicago, con el fin de combatir y eliminar el crimen organizado que asola la ciudad en estos años veinte. Pero es un trabajo difícil a pesar de la categoría de *Mike,* pues ha de enfrentarse cuerpo a cuerpo contra los cinco grandes jefes de la mafia y sus respectivos secuaces que tienen dominada y aterrorizada a toda la ciudad.

El juego consta de noventa y dos pantallas por las cuales tendrás que ir con todos los sentidos ya que en el lugar más inesperado aparecerá alguno de los grandes capos, nombre con los que se les designa a los mafiosos más importantes y sus esbirros, sin darte tiempo al menos de escapar, pero tu misión no sólo es ésta, sino que debes eliminarlos para que Chicago vuelva a ser la ciudad de antes.

Cada vez que en un barrio consigas eliminar al gran mafioso, junto con él irán desapareciendo sus esbirros y no tendrás que volver a esa zona. Para más datos te diremos una referencia de los mafiosos a los cuales debes batir: *Roddy Buldog,* jefe de los atracadores; *Jonhny Fandango,* amo y señor de los contrabandistas por herencia de familia; *Franky Frondasio,* corrupto policía que de vez en cuando hace la vista gorda; y *El Padrino,* su historia es de sobra conocida.

Otro factor que debes tener en cuenta es que no sólo recibirás el fuego enemigo de tus enemigos más cercanos, sino que en las diferentes casas que forman la ciudad se encuentran extorsionadores que tirarán cargas de dinamita sobre ti, produciéndote un estado de embriaguez, que te pondrás a dar vueltas como un loco, habrás perdido una vida.

Cuentas tan sólo con el silencio de la noche, tu intuición y una pequeña ametralladora, que será tu fiel compañera, pero las balas se acaban, sólo podrás conseguir más munición cuando mates a un esbirro, pues éste dejará su munición la cual debes recoger rápidamente y podrás continuar tu misión.

Para aquellos que no tengáis joystick, podréis utilizar algunas teclas de función del teclado, pero esto supone mayor habilidad porque los movimientos deben ser muy concisos, equilibrados y rápidos, ya que una vez se abra el fuego no tendréis un sólo momento tranquilo.

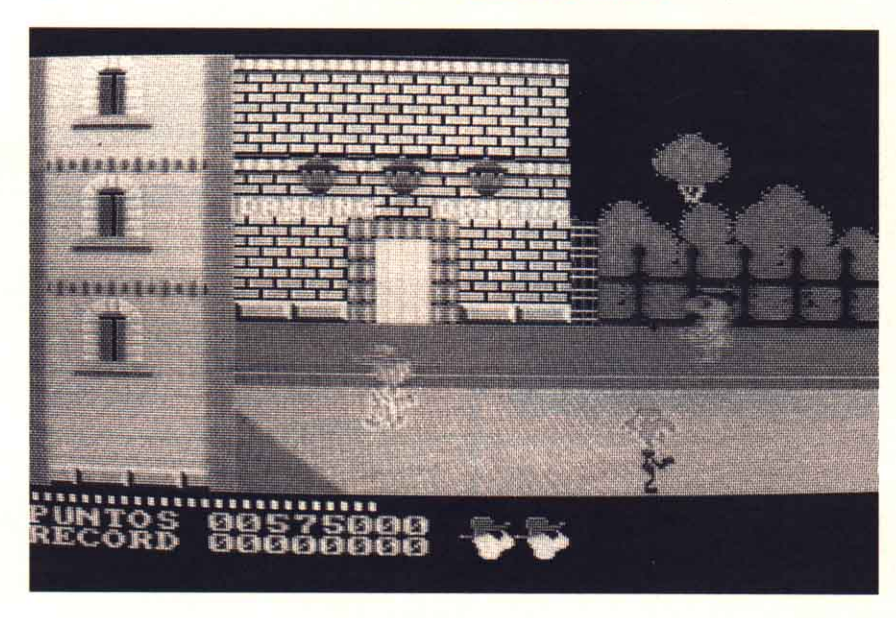
Intentad moveros por todas las avenidas de la gran ciudad y aniquilad a todos los mafiosos, os acompañará un bonito escenario, el cual nos simula los edificios, las avenidas e incluso los lugares de reunión de estos grupos tan corrompidos como ellos mismos.

Nuestro inspector *Mike Bronco,* enmarcado en ese conjunto de edificios que forman el escenario, a veces resulta un poco dificultoso de ver, ya que la indumentaria de que le han provisto son colores muy pálidos en relación al entorno y en cuanto a los esbirros la monocromía de su figura nos hace diferenciarlos más fácilmente, pero siempre se puede mejorar.

El continuo detonar de las bombas y los disparos puede hacer un poco aburrido el juego, pero esto es un elemento accesorio.

¡Animo, superdetective!, a recorrer toda la ciudad y elimina uno por uno los reyes del hampa.

Puntuación:
Presentación: 8
Claridad: 7
Rapidez: 7
Adicción: 8



Programa: Dinamite Dan
Tipo: Juego
Distribuidor: Mirrorsoft
Formato: Cassette

Este programa es otra de las creaciones que nos llegan directamente desde Inglaterra. Su presentación es sin duda alguna excelente, y su contenido, no desmerece en absoluto el conjunto del juego en sí.

Teniendo como base el polémico mundo de la guerra fría, claro está, reducido a un simple juego y sin consecuencias, *Dinamite Dan,* es la creación de un héroe moderno, que como mayor preocupación debe salvar al mudno de los designios de un loco, el *Dr. Blitzen* y su ayudante *Donna,* enemigos que nos aguardarán a lo largo del programa y que son los más peligrosos.

Llegarás al centro experimental del *Dr. Blitzen* y tienes como primera misión librarte de los astutos guardianes que son criaturas de todas las especies y los cuales no sólo te entorpecerán el camino, sino que cobrarán cada vez que caigas en sus redes una vida más. Empezarás el juego con nueve vidas.

Irás descubriendo nuevos pasadizos e irás poco a poco descubriendo dónde se encuentra el escondite del *Dr. Blitzen* y su amiga *Donna,* otra de las misiones que debes llevar a cabo, pues así no podrán volver a atemorizar al mundo.

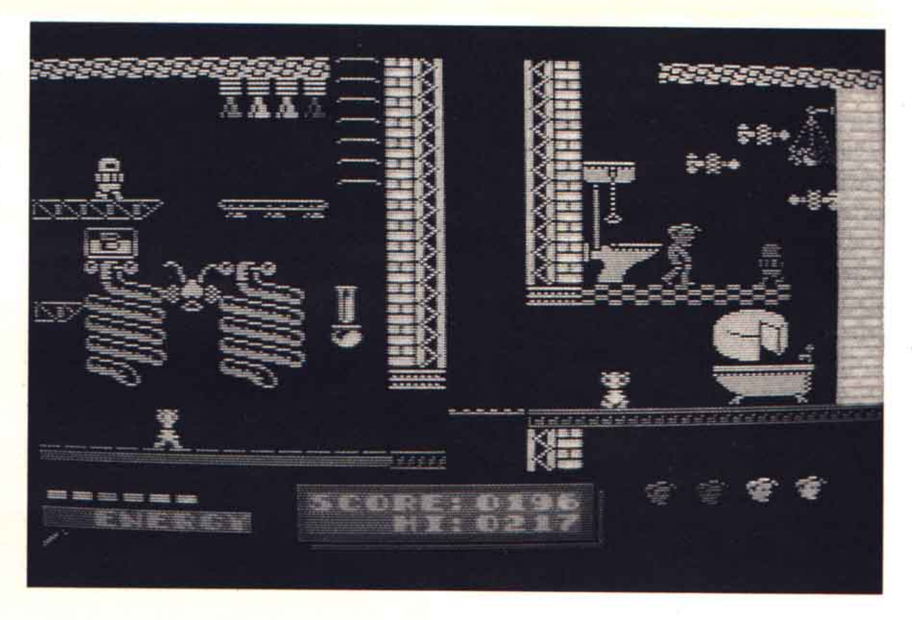
Para resistir a las continuas luchas y enfrentamientos encontrarás por los distintos bancos de pruebas, platos de alimentos y maravillosos postres que te darán la suficiente vitalidad y sumarás puntos. ¿Pero cuál es el principal objetivo de la misión?, destruir el último invento del *Dr. Blitzen,* que permitiría la dominación del mundo por una sola persona y, lo que es peor, con fines maléficos. No puedes dejar que ocurra eso, es por ello que te lanzas al ataque y buscas aprovisionarte. Las armas con las que cuentas son muy pocas pero tienes la suficiente agilidad y rapidez para esquivar todos los monstruos que se avecinan contra ti, cuando llegues a sitios a los que no puedes pasar, tendrás siete cargas de dinamita para hacerlos explosionar por los aires, y no existirá problema alguno.

Intenta subsistir el mayor tiempo posible sin utilizar estas cargas ya que al final de la aventura se encuentra el principal y más terrible problema.

Si cumples tu misión podrás escapar en el Zeppelin del Doctor, pero éste se encuentra también muy protegido por lo que no te será fácil huir, no obstante, no desesperes, es un buen juego y te hará pasar una aventura entretenida.

Nada hay que decir en cuanto a las características del programa, en el aspecto técnico, pues no es que valoremos a los profesionales ingleses, pero el resultado está a la vista.

Los *sprites* son totalmente diferentes a los que conocíamos y los elementos con los que nos sorprenden a lo largo del juego, como postres, efectos especiales y un enrevesado mundo de ficción, sin duda le permite el calificativo de un juego genial.



La acción se desarrolla rápida y la tensión es continua, a su vez es sencillo y no requiere gran esfuerzo.

Puntuación:
Presentación: 7
Claridad: 6
Rapidez: 7
Adicción: 6

SOFTWARE

Programa: Finders Keepers.
Tipo: Juego.
Distribuidor: DROSOFT.
Formato: Cassette.

Podríamos definir este juego como el espíritu de la imaginación y la fantasía reunidos.

Las directrices que marcan el juego no son espectaculares pues sigue de forma uniforme las características de otros juegos similares que podemos encontrar en el mercado, aunque esto no debe de servir para enmarcarlo en aquel tipo de juegos a los que todos hemos jugado.

Su creación gráfica, es uno de los elementos que le diferencian sumamente de los demás juegos de este tipo, ya que nuestro caballero andante y todo a su alrededor encierra una misteriosa imaginación y un cuidado trabajo en todos los elementos que forman el conjunto, color y forma de *sprites*. Mastertronic, ha cuidado en detalle, especialmente los dibujos, término que podemos emplear ya que estos parecen verdaderas caricaturas, tanto por sus movimientos como por la acción que desarrollan y su representación gráfica.

Nos trasladamos a *Isbisima,* un lugar en tu imaginación donde te ocurrirán las cosas más inverosímiles. El rey de país tiene un grave problema, es el cumpleaños de su hija y no sabe que regarlarle, te ha encargado a ti que busques un regalo digno de su hija. Para ello te tienes que trasladar al castillo de Duendilandia, donde puedes encontrar las cosas más variadas y extrañas pero a la vez más peligrosas.

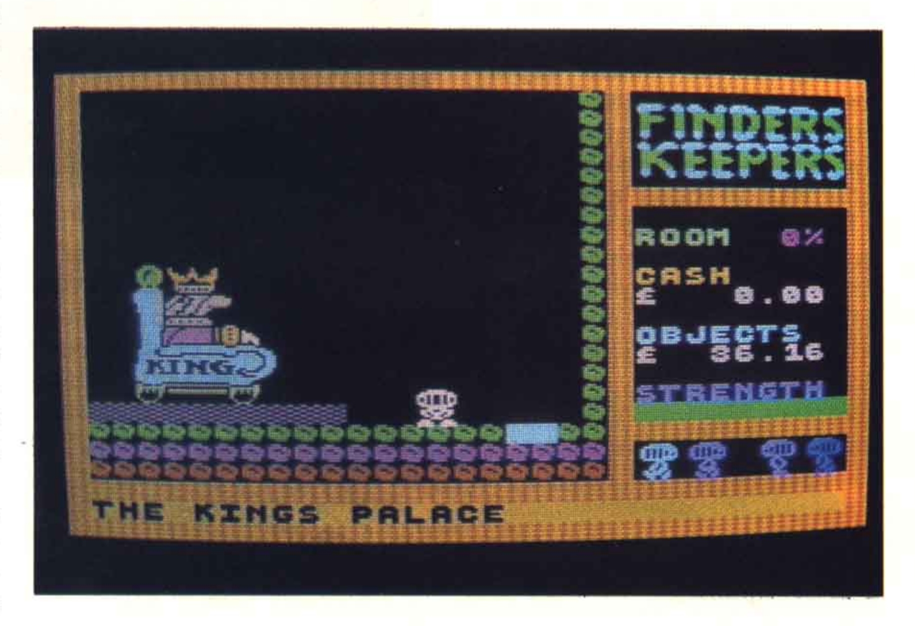
Una plétora de extraños y misteriosos mostruos saldrán a tu paso, además de las espeluznantes sanguijuelas y cocos que te introducirán en los laberintos de los que jamás podrás salir... pero el juego no consiste sólo en esto irás recogiendo toda clase de elementos que encuentres a tu paso, los cuales te servirán muchas veces para intercambiar por otros más necesarios, o te permitirán el acceso a lugares jamás habitados por el hombre.

Tendrás cuatro teclas de función designada para estos que haceres: G, para recoger objetos; T, para comerciar; D, para dejar caer y E para examinar, ya que esto es imprescindible no vaya a ser que transportes algo que no sea necesario.

Tienes dos opciones a lo largo del juego o bien intentar conseguir el mayor número de tesoros y escăparte con ellos o entregárselos al rey y entrar a formar parte de la esperada Tabla Poligonal, y entrar al servicio de su Majestad con los más distinguidos honores.

Debes dedicar la mayor parte de tu tiempo a recorrer todas las salas pues cuantos más objetos encuentres más posibilidades tendrás de salir con vida de la misión. Enfréntate a los monstruos tan sólo cuando sea necesario y no pierdas nunca la calma.

La misión ya ha sido cumplida y el principal objetivo también, pasar un largo y entretenido rato. Veréis que a lo largo del programa os acompaña una gradable música, o sintonía del juego que os divertirá y hace al programa más singular.



Puntuación:
Presentación: 8
Claridad: 7
Rapidez: 7
Adicción: 8

Programa: Fernando Martín.
Tipo: Juego.
Distribuidor: DINAMIC.
Formato: Cassette.

Tenemos ante nosotros un programa de creación *made in Spain,* el cual no es que sea una sublimación en el mundo del *software* pero es una contribución interesante y agradable por el tema que trata, el cual ya habréis imaginado.

El baloncesto, tanto en España como en el mundo entero es un deporte que está haciendo furor entre la gente, incluso como espectáculo. La NBA, es una liga de baloncesto en la que tan sólo 264 hombres pueden jugar y entre ellos se encuentra una de las estrellas favoritas de nuestro baloncesto, Fernando Martín, a quien va dedicado por entero este programa.

Podremos disfrutar de un interesante encuentro de habilidad, rapidez y preparación, si optamos medir nuestras fuerzas frente a este coloso del baloncesto.

Cuando salgas a la cancha te encontrarás nervioso, preocupado y con deseos de hacer un buen partido, pero al ver quién es tu adversario..., las cosas cambian, claro que no es necesario que adoptes esta opción de juego, también pueden ser dos personas las que jueguen entre ellos, con la particularidad, de que siendo tan personal el campo de juego y las técnicas puedas jugar bien con cursores, *joystick,* o teclado.

¿Qué cosas puedes hacer en la cancha?, frente a «Martín», pocas cosas te podemos decir pero no te desanimes.

Existen tiros de tres puntos, robo de balones y los tan de moda mates, de espalda, de lado o como mejor prefieras, todo ello unido a que el juego se puede desarrollar libremente pues el árbrito no suele estar muy atento a los golpes bajos, y tienes que estar siempre dispuesto a recoger los rebotes porque sino pasarás la partida sin enterarte.

Consta de dos tiempos de duración, cinco minutos cada uno, excesivo a nuestro parecer, pues podría existir una opción que delimitase el tiempo. Lento, en toda la extensión de la palabra, cuando nuestro ídolo lleva la pelota nos parece imposible quitársela y además los dobles son facilísimos de hacer debido a la lentitud del salto, no se coordinan bien los movimientos.

Carece completamente de sonido, tan sólo el aplauso del público que rompe un poco la monotonía de la acción y el cual resulta a veces innecesario.

Color, se ve reflejado tan sólo en los marcadores, ya que la cancha y los jugadores sólo representan los colores blanco y negro como base, no sabemos a qué puede ser debido esto, la única razón que se nos ocurre es para visualizar mejor los movimientos, pero eso lo dejamos a vuestra crítica.

Lo principal en este juego, es familiarizarse lo mejor posible con los mandos que emplees y podréis realizar jugadas y acciones bonitas, todo es intentarlo.

Debemos elogiar la caricatura de Fernando Martín que aparece el inicio del programa y en la que se encuentra muy favorecido, es la realidad llevada al ordenador.

Unas dosis de entrenamiento y una gran ilusión os puede hacer llegar como a F. Martín donde nunca antes nadie había llegado.

Puntuación:
Presentación: 6
Claridad: 7
Rapidez: 6
Adicción: 7

MSX SERVICIO



Núm. 1
¿Qué es el MSX? Su BASIC, periféricos, programas, software.



Núm. 2
Generación de sonido. MSX-DOS, el ordenador por dentro, programas, noticias.



Núm. 3
Los joysticks, 256 caracteres programables, Z80 corazón de león, compro/vendo/cambio.



Núm. 4
Las comunicaciones entre ordenadores, la jerga informática, trucos,



Núm. 5
Comandos de entrada/salida, el BASIC MSX comparado con Spectrum y Commodore 64. Código Máquina.



Núm. 6
Los 8 magníficos (test gigante), el bus de expansión, los misterios de la grabación, programas.



Núm. 7
Analizamos el Generador de Sonido. Aplicaciones matemáticas con el ordenador.



Núm. 8
Compact Disc. el periférico del futuro. Test: Dynadata DPC-200. Continuamos con la memoria de vídeo. Libros,



Núm. 9
Características técnicas del Compact Disc. Tratamiento de datos. Test: Quick Disk. Trucos, libros, noticias,



Núm. 10
Características de la II Generación. Los secretos del modo Screen 2. Test: los plotters. Aplicaciones: matrices y determinantes.



Núm. 11
LOGO, un lenguaje educativo. Screen 3: el modo multicolor. Aplicaciones: sistemas de ecuaciones. BASIC para principiantes. Test: Seikosha SP-1000MX.



Núm. 12
SVI-328: precursor del estándar. Aplicaciones: sistemas de ecuaciones II. Código Máquina. Test: Toshiba HX-20.

DE EJEMPLARES ATRASADOS

ESTOS SON LOS EJEMPLARES DE MSX MAGAZINE APARECIDOS EN EL MERCADO CON UN RESUMEN DE SU CONTENIDO



Núm. 13
VG-8235, la I generación en marcha. SVI-318/328: análisis interno. Test: Yamaha CX5M y CX5M II. BASIC: las variables alfanuméricas. Las matemáticas y el ordenador.



Núm. 14
Controle sus errores de programación. Aplicaciones matemáticas: interpolación. Memoria de Vídeo: los sprites. Código Máquina: los registros dobles.



Núm. 15
¿Porqué es lento el BASIC? El procesador de vídeo del SVI-318/328. Test: Sony HB-500P. BASIC: los diagramas de flujo. Los modos de pantalla.



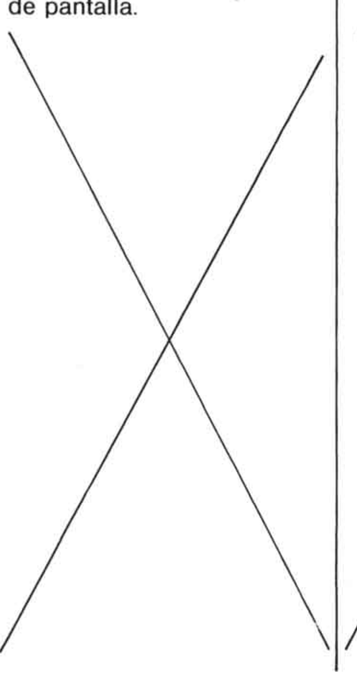
Núm. 16
Dos gigantes frente a frente. Test: VC-10, un osciloscopio muy especial. Síntesis de voz. Utilidades de la RAM. Memoria de vídeo: instrucciones VPEED y VPOKE.



Núm. 17
Robots, trabajadores infatigables. Cómo ahorrar memoria. Test: Mitsubishi ML-G1 y ML-G3. Instrucciones ocultas del Z-80. El procesador de vído del SVI-318/328. Desensamblador.



Núm. 18
Los diskettes al descubierto. El BIOS de la memoria de vídeo. Test: interface RS-232C. Unidad de discos ML-F30D. Utilización de ficheros. SVI-318/328, SCREEN 2.



PARA HACER SU PEDIDO, RELLENE ESTE CUPON, HOY MISMO Y ENVIELO A **MSX MAGAZINE**
BRAVO MURILLO, 377. Tel. 7337969 - 28020 MADRID

Ruego me envíen los siguientes números atrasados de MSX ____________________
al precio de **300** ptas. cada uno. Cuyo importe abonaré:
☐ POR CHEQUE ☐ CONTRA REEMBOLSO ☐ CON MI TARJETA DE CREDITO
☐ AMERICAN EXPRESS ☐ VISA ☐ INTERBANK

Número de mi tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
Fecha de caducidad ____________________
NOMBRE ____________________
DIRECCION ____________________
POBLACION ____________________ C.P. __________
PROVINCIA ____________________

Estática de fluidos



El programa muestra hasta qué punto el ordenador puede convertirse en un compañero más, a la hora de resolver los complicados cálculos físicos que plantea esta rama de la ciencia.

Se trata de un programa educativo que nos enseña cuatro ecuaciones básicas de la estática de fluidos, permitiéndonos operar con ellas. Estas cuatro ecuaciones son las siguientes:

1. Ecuación fundamental de la estática de fluidos.
2. Principio de la prensa hidráulica.
3. Principio de los vasos comunicantes.
4. Principio de Arquímedes.

El programa está pensado para que la aplicación práctica de las ecuaciones se realice utilizando el sistema internacional de unidades o M.K.S., y por ello trae un apartado dedicado al cambio de unidades.

La presentación en pantalla está bien y facilita el entendimiento de las dintintas fórmulas y su aplicación.

Rodolfo Müller
Cádiz

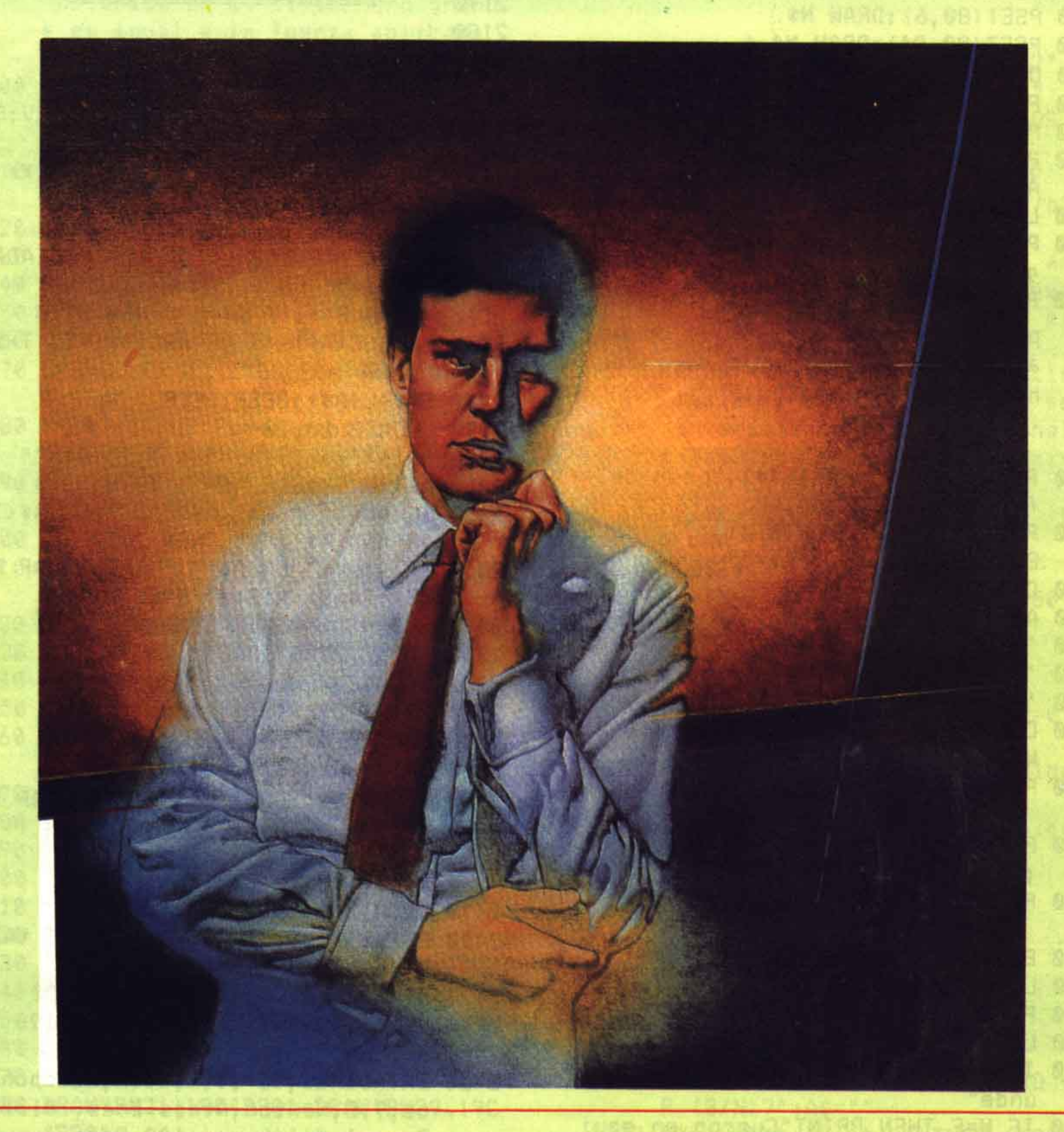
```
10 '*******************************
20 '*                             *
30 '*    ESTATICA DE FLUIDOS      *
40 '*                             *
50 '*           POR:              *
60 '*                             *
70 '*      RODOLFO MUELLER        *
80 '*                             *
90 '*******************************
100 '
110 '     PRESENTACION Y MENU
120 '     ===================
130 '
140 CLS:COLOR 4,1,1:SCREEN 3
150 OPEN "GRP:" FOR OUTPUT AS#1
160 PRESET(10,20):PRINT#1,"ESTATICA"
170 PRESET(100,85):PRINT#1,"DE"
180 PRESET(30,150):PRINT#1,"FLUIDOS"
190 FOR X=1 TO 1500:NEXT X
200 CLS:PRESET(20,10):PRINT#1,"Use el
    "
210 PRESET(20,85):PRINT#1,"sistema"
220 PRESET(40,150):COLOR 12:PRINT#1,"
    M.K.S."
230 FOR X=1 TO 2500:NEXT X
240 CLOSE#1
250 M$="C1U20L3E3F3L3":N$="C1D20L3F3E
    3L3"
260 DIM R$(7),N(7),T$(7)
270 FOR X=1 TO 7
280 READ R$(X),N(X),T$(X)
290 NEXT X
300 CLS:COLOR 12,1:SCREEN 0:KEY OFF
310 PRINT:PRINT
```

```
320 PRINT"    TECLA          OPCION"
330 PRINT"    ======================
    ======":PRINT
340 PRINT"      1    Ecuación fundame
    ntal":PRINT
350 PRINT"      2    Prensa hidraulic
    a":PRINT
360 PRINT"      3    Vasos comunicant
    es":PRINT
370 PRINT"      4    Ppio. de Arquime
    des":PRINT
380 PRINT"      5    Conversión de un
    idades":PRINT
390 LOCATE 3,19
400 PRINT">>> PULSE LA OPCION DESEADA
     <<<"
410 A$=""
420 A$=INKEY$:IF A$="" THEN 420
430 IF ASC(A$)<48 OR ASC(A$)>53 THEN
    420
440 A=VAL(A$)
450 ON A GOSUB 510,910,1340,1790,2190
460 GOTO 300
470 '
480 '      EC. FUNDAMENTAL
490 '      ===============
500 '
510 CLS:COLOR 1,12,12:SCREEN 2
520 OPEN "GRP:" FOR OUTPUT AS#1
530 LINE(21,30)-(139,99),4,BF
540 DRAW"BM21,27C4D3":DRAW"BM139,27C4
    D3"
550 PSET(80,6):DRAW N$
560 LINE(110,30)-(110,80)
570 LINE(108,32)-(112,28)
580 LINE(108,82)-(112,78)
590 CIRCLE(50,80),1,1,,,1.4
600 PRESET(85,12):PRINT#1,"P"
610 PRESET(60,75):PRINT#1,"P'"
620 PRESET(115,50):PRINT#1,"h"
630 PRESET(120,10):PRINT#1,"EC. FUNDA
    MENTAL"
640 LINE(150,50)-(250,78),15,BF
650 PRESET(160,60):PRINT#1,"P'=P+δ*g*
    h"
660 PRESET(8,110):PRINT#1,"La presión
     en un punto del lí-  quido es i
    gual a la presión P   en la supe
    rficie más el peso de"
670 PRESET(8,134):PRINT#1,"una column
    a liquida que tenga   por base l
    a unidad de superfi-  cie y por
    altura la distancia"
680 PRESET (8,158):PRINT#1,"vertical
    entre dicho punto y la superfici
    e libre."
690 PRESET (30,185):PRINT#1,">>> PULS
    E UNA TECLA <<<":A$=""
700 CLOSE#1
710 A$=INKEY$:IF A$="" THEN 710
720 '
730 '      ================
740 '
750 CLS:COLOR 12,1:SCREEN 0:P1=0:D=0:
    H=0
760 PRINT:INPUT"Presión en superficie
    ;   P=";P1
770 PRINT:INPUT"Densidad del líquido;
        δ=";D
780 PRINT:INPUT"Altura punto-superfic
    ie; h=";H
790 P2=P1+D*H*9.8:P=CSNG(P2)
800 LOCATE10,12
810 PRINT"P'=";P;" Pascal"
820 LOCATE 0,20
830 PRINT"- DESEA CALCULAR OTRO VALOR
      (S/N)? -":A$=""
840 A$=INKEY$:IF A$="" THEN 840
850 IF A$="S" OR A$="s" THEN 750
860 RETURN
870 '
880 '      PRENSA HIDRAULICA
890 '      =================
900 '
910 CLS:COLOR 1,14,14:SCREEN 2
920 LINE(21,36)-(39,99),4,BF
930 LINE-(149,81),4,BF
940 LINE-(101,36),4,BF
950 DRAW"BM21,29C4D3":DRAW"BM39,29C4D
    3":DRAW"BM149,29C4D3":DRAW"BM101
    ,29C4D3"
960 LINE(21,32)-(39,35),1,BF
970 LINE(101,32)-(149,35),1,BF
980 LINE(110,20)-(140,31),8,BF
990 PSET(30,8):DRAW N$
1000 PSET(125,60):DRAW M$
1010 OPEN "GRP:" FOR OUTPUT AS#1
1020 PRESET(35,15):PRINT#1,"f"
```

```
1030 PRESET(45,30):PRINT#1,"s"
1040 PRESET(130,45):PRINT#1,"F"
1050 PRESET(155,30):PRINT#1,"S"
1060 PRESET(100,8):PRINT#1,"PRENSA HI
     DRAULICA"
1070 LINE(170,50)-(246,78),12,BF
1080 PRESET(180,60):PRINT#1,"F=f*S/s"
1090 PRESET(8,110):PRINT#1,"La fuerza
      obtenida en el cilin- dro grand
     e es igual a la fuerza aplicada
     al cilindro pequeño,"
1100 PRESET(8,134):PRINT#1,"multiplic
     ada por la relación    entre las
      areas de los dos ém-  bolos."
1110 PRESET(30,185):PRINT#1,">>> PULS
     E UNA TECLA <<<":A$=""
1120 CLOSE#1
1130 A$=INKEY$:IF A$="" THEN 1130
1140 '
1150 '     ================
1160 '
1170 CLS:COLOR 12,1:SCREEN 0:F2=0:S2=
     0:S1=0
1180 PRINT:INPUT"Fuerza aplicada;
        f=";F2
1190 PRINT:INPUT"Area del émbolo meno
     r; s=";S2
1200 IF S2=0 THEN 1170
1210 PRINT:INPUT"Area del émbolo mayo
     r; S=";S1
1220 F1=F2*S1/S2:F=CSNG(F1)
1230 LOCATE 10,12
1240 PRINT"F=";F;" Newton"
1250 LOCATE 0,20
1260 PRINT"- DESEA CALCULAR OTRO VALO
     R (S/N)? -":A$=""
1270 A$=INKEY$:IF A$="" THEN 1270
1280 IF A$="S" OR A$="s" THEN 1170
1290 RETURN
1300 '
1310 '     VASOS COMUNICANTES
1320 '     ==================
1330 '
1340 CLS:COLOR 1,14,14:SCREEN 2
1350 LINE(21,50)-(39,99),4,BF
1360 LINE-(129,81),4,BF
1370 LINE-(111,65),4,BF
1380 DRAW"BM21,19C4D30":DRAW"BM39,19C
     4D30"
1390 LINE(111,65)-(129,22),12,BF
1400 DRAW"BM129,19C4D46":DRAW"BM111,1
     9C4D46"
1410 DRAW"BM45,50C4D15R60U43"
1420 LINE(43,52)-(47,48),4
1430 LINE(43,67)-(47,63),4
1440 LINE(103,24)-(107,20),4
1450 LINE(103,67)-(107,63),4
1460 OPEN "GRP:" FOR OUTPUT AS#1
1470 PRESET(50,53):PRINT#1,"h2"
1480 PRESET(85,38):PRINT#1,"h1"
1490 PRESET(70,86),4:PRINT#1,"δ2"
1500 PRESET(113,38):PRINT#1,"δ1"
1510 PRESET(100,2):PRINT#1,"VASOS COM
     UNICANTES"
1520 LINE(140,40)-(248,68),10,BF
1530 PRESET(150,50):PRINT#1,"δ1/δ2=h2
     /h1"
1540 PRESET (8,110):PRINT#1,"Si ambos
      vasos poséen el mismo  líquido
     el nivel alcanzado se-  rá el mi
     smo.Si hay dos líquidos"
1550 PRESET(8,134):PRINT#1,"distintos
      las alturas alcanza-  das serán
      inversamente propor-  cionales
     a las densidades res-  pectivas.
     "
1560 PRESET(30,185):PRINT#1,">>> PULS
     E UNA TECLA <<<":A$=""
1570 CLOSE#1
1580 A$=INKEY$:IF A$="" THEN 1580
1590 '
1600 '     =============
1610 '
1620 CLS:COLOR 12,1:SCREEN 0:D1=0:D2=
     0:H1=0
1630 PRINT:INPUT" Densidad líquido 1;
      δ1=";D1
1640 PRINT:INPUT" Densidad líquido 2;
      δ2=";D2
1650 IF D2=0 THEN 1620
1660 PRINT:INPUT" Altura 1;
      h1=";H1
1670 H2=H1*D1/D2:H=CSNG(H2)
1680 LOCATE 10,12
1690 PRINT"h2=";H;" Metros"
1700 LOCATE 0,20
1710 PRINT"- DESEA CALCULAR OTRO VALO
     R (S/N)?":A$=""
1720 A$=INKEY$:IF A$="" THEN 1720
1730 IF A$="S" OR A$="s" THEN 1620
1740 RETURN
1750 '
```

```
1760 '     PPIO. DE ARQUIMEDES
1770 '     ===================
1780 '
1790 CLS:COLOR 1,12,12:SCREEN 2
1800 LINE(21,42)-(139,99),4,BF
1810 LINE(50,30)-(110,60),8,BF
1820 DRAW"BM21,39C4D3":DRAW"BM139,39C
     4D3"
1830 PSET(80,6):DRAW N$
1840 PSET(80,84):DRAW M$
1850 OPEN "GRP:" FOR OUTPUT AS#1
1860 PRESET(85,15):PRINT#1,"W"
1870 PRESET(85,68):PRINT#1,"E"
1880 PRESET(102,2):PRINT#1,"PPIO. DE
     ARQUIMEDES"
1890 LINE(155,50)-(247,78),14,BF
1900 PRESET(165,60):PRINT#1,"E=V1*δl*
     g"
1910 PRESET(8,118):PRINT#1,"Todo cuer
     po sumergido en un lí- quido est
     á sometido a un empuje dirigido
     hacia arriba igual al"
1920 PRESET(8,142):PRINT#1,"peso del
     fluido desalojado."
1930 PRESET(30,185):PRINT#1,">>> PULS
     E UNA TECLA <<<":A$=""
1940 CLOSE#1
1950 A$=INKEY$:IF A$="" THEN 1950
1960 '
1970 '      ==============
1980 '
1990 CLS:COLOR 12,1:SCREEN 0:V=0:D=0:
     W=0
2000 PRINT:INPUT"Volumen desalojado;
       V1=";V
2010 PRINT:INPUT"Densidad del líquido
     ; δl=";D
2020 PRINT:INPUT"Peso del cuerpo;
       W=";W
2030 E=V*D*9.8:E=CSNG(E):W=CSNG(W)
2040 LOCATE 10,12
2050 PRINT"E=";E;" Newton"
2060 LOCATE 10,15
2070 IF W>E THEN PRINT"El cuerpo se h
     unde"
2080 IF W=E THEN PRINT"Cuerpo en equi
     librio"
2090 IF W<E THEN PRINT"El cuerpo flot
     a"
2100 LOCATE 0,20
2110 PRINT"- DESEA CALCULAR OTRO VALO
     R (S/N)? -":A$=""
2120 A$=INKEY$:IF A$="" THEN 2120
2130 IF A$="S" OR A$="s" THEN 1990
2140 RETURN
2150 '
2160 '    CONVERSION DE UNIDADES
2170 '    ======================
2180 '
2190 CLS:COLOR 5,1:V=0
2200 FOR X=1 TO 7:V=V+2
2210 LOCATE 2,V:PRINT X:LOCATE 7,V:PR
     INT R$(X)
2220 LOCATE 19,V:PRINT"<<>> ";T$(X)
2230 NEXT X
2240 LOCATE 5,18
2250 PRINT"- PULSE LA OPCION DESEADA
     -":A$=""
2260 A$=INKEY$:IF A$="" THEN 2260
2270 IF ASC(A$)<49 OR ASC(A$)>55 THEN
      2260
2280 S=VAL(A$):BEEP:BEEP
2290 LOCATE 10,20
2300 PRINT"sentido  < ó > ?":A$=""
2310 A$=INKEY$:IF A$="" THEN 2310
2320 IF A$<>"<" AND A$<>"," AND A$<>"
     >" AND A$<>"." THEN 2310
2330 IF A$="<" OR A$="," THEN SWAP R$
     (S),T$(S):N(S)=1/N(S)
2340 CLS:N(S)=CSNG(N(S)):C=0
2350 INPUT "CANTIDAD:";C
2360 LOCATE 4,9
2370 PRINT C;R$(S);" son:"
2380 LOCATE 4,11
2390 PRINT C*N(S);T$(S)
2400 IF A$="<" OR A$="," THEN SWAP R$
     (S),T$(S):N(S)=1/N(S)
2410 LOCATE 0,20
2420 PRINT"- DESEA CALCULAR OTRO VALO
     R (S/N)? -":A$=""
2430 A$=INKEY$:IF A$="" THEN 2430
2440 IF A$="S" OR A$="s" THEN 2190
2450 RETURN
2460 DATADinas,1E-05,Newton,Kilopondi
     os,9.8,Newton,Atmósferas,101325,
     Pascal,Milibares,100,Pascal
2470 DATATorr,113.32,Pascal,Litros,
     1E-03,m^3,grs./c.c.,1000,Kg./m^3
```

Aplicaciones matemáticas



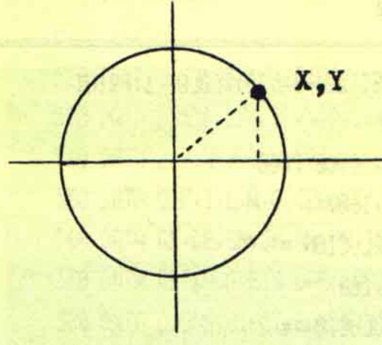
Aunque las coordenadas rectangulares tienen una representación algo rígida en las definiciones de 256x192 al no coincidir los *pixel's* del código entre la salida MSX y los televisores encargados de visualizarlos, consideramos útil el análisis de este caso para el aprendizaje y algún otro problema que pueda surgirnos.

Asi: (ver figura 1)

aplicamos la fórmula que relaciona las coordenadas rectangulares entre sí y con respecto al radio de una circunferencia:

$X = r^2 - y^2 \qquad Y = r^2 - x^2$

y su representación geométrica:



Los bucles 60-90 y 100-130 indican los puntos relativos para iniciar y finalizar la circunferencia por este método.

```
MODELO 2πR

        10 CLS
   20  INPUT "RADIO";R
   30  INPUT "ABCISA";X
   40  INPUT "ORDENADA";Y
   50  SCREEN 2
   60  FOR I = X-R TO X+R STEP .5
   70  J=YSQR((R^2)-((X-I)^2))
   80  PSET (I,J)
   90  NEXT
  100  FOR I=X+R TO X-R STEP -.5
  110  J=Y+SQR((R^2)-((X-I)^2))
  120  PSET (I,J)
  130  NEXT
  135  CIRCLE (X,Y), R+20, 5
  140  GOTO 140
```

Figura 1

La línea 135 dibuja una tenue circunferencia por esterografía para permitir la comparación de resultados elocuentemente entre el desarrollo propuesto y la aplicación de la instrucción *CIRCLE*.

La lentitud e imperfección no son óbice para aplicar a ciertas utilidades gráficas de arcos, por ejemplo, rotulaciones y otras como ya veremos. (Ver figura 2).

NOTA: Advertimos que en las líneas 80, 110, 120 y 140 se maneja la variable O a la que no hay que confundir con el cero = 0.

Este ejemplo aplica las razones trigonométricas por medio de las relaciones en coordenadas polares, líneas 110 y 120. Por lo demás es muy parecido al modelo 2μR.

En la línea 130 se resuelve la repetición para *PSET* por medio de un bucle artificial entre 100 y 150 en lugar del conocido *FOR/next*.

En la 140 la variable V aumenta un paso cada vez hasta igualar a P, es decir, los puntos definidos (máximo) que indica su término.

En aplicaciones gráficas puede resultar interesante el origen y

```
MODELO TRIG

   10 CLS
   20 DEFSNG A-Z
   30 INPUT "nº PUNTOS DEFINICION"; P
   40 INPUT "RADIO"; R
   50 PRINT "COORDENADA"
   60 INPUT "X"; X
   70 INPUT "Y"; Y
   80 O=0: IO=2*3.14/P
   90 SCREEN 2
  100 IF V=P THEN 160
  110 AF=X+R COS(O)
  120 OF=Y+R SIN(O)
  130 PSET(AF,OF)
  140 O=O+IO: V=V+1
  150 GOTO 100
  160 GOTO 160
```

Figura 2

```
10 CLS:COLOR 10,1,1
20 X=128:Y=80:R=80
30 A=0:B=2*3.1415/6:W=90
40 OPEN"GRP:"AS#1
50 SCREEN 2
60 GOSUB 5000
70 N=N+1:EA=0:EB=0:EC=0:ED=0:EE=0:EF=0
75 IF N=>17 THEN 400
80 ON N GOSUB 1000,1100-1200,1300,1400,1
500,1600,1700,1800,1900,2000,2100,2200,2
300,2400,2500
83 IF N>9 THEN IX=4 ELSE IX=0
84 CIRCLE(230,32),15,2:PAINT STEP(0,0),2
:PLAY"O7T250S8L50M20A"
85 COLOR 2:PSET(220-IX,30):COLOR 1:PRINT
#1,N
86 COLOR 2:PSET(221-IX,30):COLOR 1:PRINT
#1,N:COLOR 10
90 GOSUB 6000:GOSUB 7000
92 PRESET(155,160)
93 PRINT#1,"SOLUCION =  "
95 PRESET(157,160)
98 PRINT#1,"SOLUCIO  = ?"
100 GOSUB700
200 CK=0:CT=0:TT=0:GOSUB 8600:PRESET(10,
160):PRINT#1,"EXACTO";"=";M$(6):GOSUB 40
00
300 GOSUB 8900:GOSUB 8800:GOSUB 8000:GOT
O 70
400 GOTO 9000
500 DATA 1,2,3,4,5,6,1000,1
510 DATA 1,2,1,1,1,2,3000,1
```

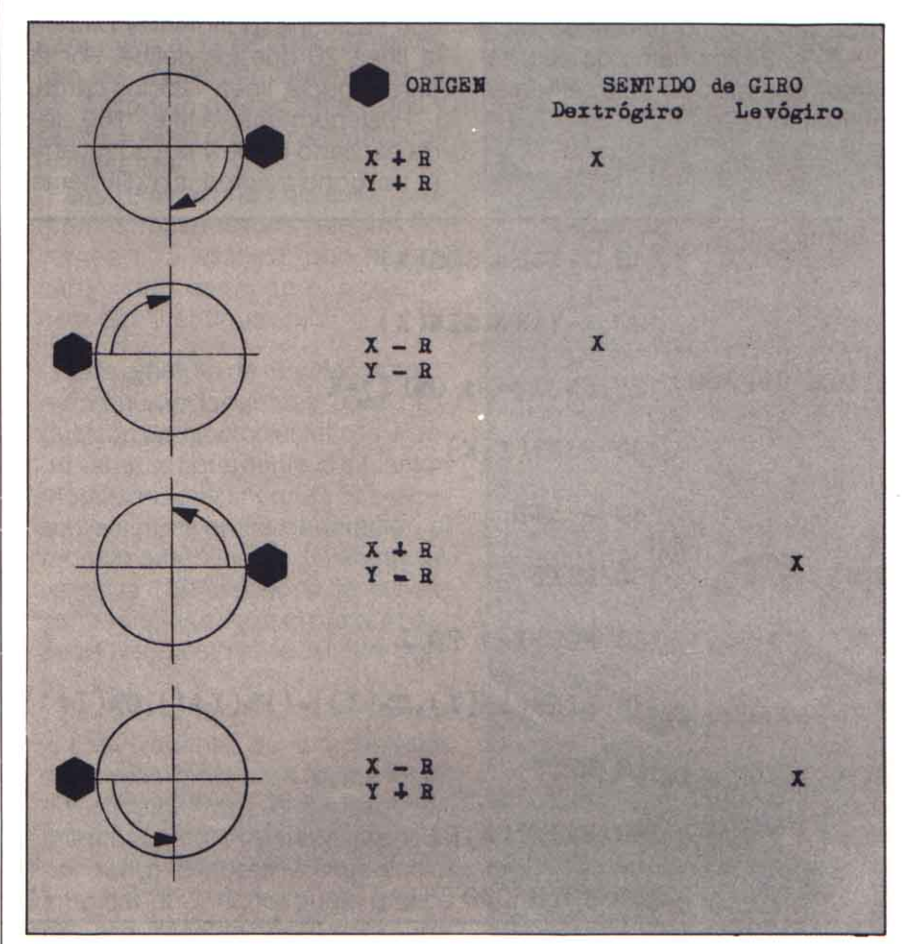


Figura 3

sentido de giro. Véase el esquema siguiente con el sentido de giro de las agujas del reloj; las dos primeras; y las dos últimas en el contrario. (Ver figura 3).

Sustituyendo en las líneas 110 y 120 las relaciones X − R por las correspondientes en esta figura se obtendrán los inicios y sentidos de giro deseados.

Aprovechando un proceso similar vamos a efectuar un análisis de su aplicación a los polígonos regulares. En este caso la calidad de los gráficos es deficiente, pero variando la precisión de PI y con un tamaño adecuado de cada gráfico y rellenándolo las figuras resultarán aceptables: (Ver figura 4).

Salvo ligeras variantes este modelo es similar al anterior. Se solicitan la coordenada y el radio y se añade el número de lados que se deseen.

Por medio del primer bucle de la línea 100 a la 150 se sitúan los puntos correspondientes a los vértices del polígono deseado.

```
520 DATA 1,2,1,3,1,4,3500,1
530 DATA 65536,256,16,4,2,1.41,4000,4
540 DATA Z,Y,X,W,V,U,1000,1
550 DATA C,F,I,L,Ñ,Q,3000,1
560 DATA AB,DE,GH,JK,MN,OP,3000,2
570 DATA AEI,OUA,EIO,UAE,IOU,AEI,4000,3
580 DATA D,ONQ,UIJOT,EDELA,MAN,C,6000,1
590 DATA 2,B,4,D,6,F,1500,1
600 DATA 27,20,13,Z,S,M,3000,1
610 DATA ROMA,OMAR,MARO,AROM,ROMA,OMAR,3
000,4
620 DATA 05,15,45,135,405,1215,4000,4
630 DATA 1X,X1,11X,X11,111X,X111,3000,4
640 DATA A(1),B(2),Y(26),Z(27),W(51),X(5
2),3500,5
650 DATA 15,1111,F-16,10000,10,4000,2
```

```
700 GOSUB 3600:CE=CE+1
710 R$=INKEY$
720 CT=CT+1:IF CT=VAL(M$(7)) THEN CT=0:G
OTO 970
730 IFR$=""THEN710ELSET$=R$:GOTO890
800 CK=CK+1:IFC<>3 THEN CK=0:SP=SP-1:GOT
O 700
805 PRESET(10,160):PRINT#1,"NO◇REPITA"
:PLAY"O8T250S14L50M25G":FORT=1TO600:NEXT
T
810 GOTO 700
890 IF LEN(T$)=LEN(M$(6))AND T$<>M$(6)TH
EN 800
900 IF T$=M$(6) THEN 200 ELSE 910
910 R$=INKEY$
920 CT=CT+1:IF CT=VAL(M$(7)) THEN CT=0:G
```

Como después se trazan los lados, se toma el número L en la línea 100 hasta un punto más (L+1) como último trazo. El bucle 100 a 150 indica, pues los puntos necesarios y el 160 a 180 instruye cada lado como una línea recta *(LINE)* entre los datos de coordenadas que previamente se han almacenado en:

F%(200) y G%(200)

con sus correspondientes *DIM* en la línea 20 que los define. Por el primer bucle, línea 125, los carga.

Posteriormente la línea 180 cierra el citado bucle y la 190 rellena el polígono trazado, simplemente

```
MODELO POLIG
10 CLS
20 DEFSNG A-Z: DIM F%(200): DIM G%(200)
30 INPUT "Nº de LADOS"; L
40 INPUT "RADIO"; R
50 PRINT "COORDENADA CENTRO"
60 INPUT "X"; X
70 INPUT "Y"; Y
80 A=0: B=2*3.14/L
90 SCREEN 2
100 FOR I=1 TO L+1
110 J=X+R*COS(A)
120 K=Y+R*SIN(A)
125 F%(I)=J: G%(I)=K
130 PSET(J,K)
140 A=A+B
150 NEXT
160 FOR I=1 TO L
170 LINE(F%(I),G%(I))-(F%(I+1),G%(I+1))
180 NEXT
190 PAINT(X,Y)
200 GOTO 200
```

Figura 4

```
0TO  970
930 IFR$=""THEN910ELSET$=T$+R$:GOTO890
970 PLAY"T255O7DGDGDG"
975 PSET(10,10):COLOR 1:PRINT#1," TIEMPO
 "
980 CT=0:TT=TT+1:IF TT= 6 THEN TT=0:GOTO
 300
982 FORT=1TO150:NEXTT
985 PRESET(5,10):COLOR 15:PRINT#1,"+++++
++++"
986 PRESET(5, 8):COLOR 15:PRINT#1,"+++++
++++"
990 GOTO 975
1000 RESTORE 500:RETURN
1100 RESTORE 510:RETURN
1200 RESTORE 520:RETURN
1300 RESTORE 530:RETURN
1400 RESTORE 540:RETURN
1500 RESTORE 550:RETURN
1600 RESTORE 560:RETURN
1700 RESTORE 570:RETURN
1800 RESTORE 580:RETURN
1900 RESTORE 590:RETURN
2000 RESTORE 600:RETURN
2100 RESTORE 610:RETURN
2200 RESTORE 620:RETURN
2300 RESTORE 630:RETURN
2400 RESTORE 640:RETURN
2500 RESTORE 650:RETURN
3000 REM
3010 REM
3050 IF LEN(M$(1))>2 THEN EA=-24
```

para mejorar la estética. Si se varían las líneas convenientemente se obtendrán diversos aspectos y se efectuarán procesos didácticos interesantes.

Las derivaciones de estos programas matemáticos pueden llevarnos a la estética, que hemos comentado como en el siguiente ejemplo: (ver figura 5).

Este ejemplo de círculos aproximadamente tangentes para producción de efectos estéticos. Puede variarse libremente la situación modificando la línea 50. El proceso es similar en su generación al modelo antecedente. La línea clave es la 160 creadora de las circunferencias según el número deseado expuesto en la 30. (Ver figura 6).

Esta variante de los modelos anteriores sirve para rotular letras en circulo. Para otros procedimientos sería preferible predecir un nuevo alfabeto y símbolos o emplear el *GML* (lenguaje gráfico de alta resolución). Podía emplearse también en *SCREEN 3.*

```
MODELO  CIRTAN
   10 CLS
   20 DEFSNG A - Z: DIM F%(200), G%(200)
   30 INPUT "TOTAL CIRCUNFERENCIAS"; L
   40 INPUT "RADIO CENTRAL"; R
   50 X = 128: Y = 90
   60 A = 0: B = 2*3.14/L: W = 360/L
   70 SCREEN 2
   80 FOR I = 1 TO L
   90 J = X + R COS (A)
  100 K = Y + R SIN(A)
  110 F%(I)= J: G%(I)= K
  120 PSET (J,K)
  130 A = A + B
  140 NEXT
  150 FOR I = 1 TO L
  160 CIRCLE (F%(I),G%(I)), R/(L 2/6)
  170 NEXT
  180 GOTO 180
```

Figura 5

```
3060 IF LEN(M$(2))>2 THEN EB=-24
3070 IF LEN(M$(3))>2 THEN EC=-12:EF=20
3080 IF LEN(M$(4))>2 THEN ED=-24
3090 IF LEN(M$(5))>2 THEN EE=-24
3100 RETURN
4000 PRESET(10,180):PRINT#1,"PARA SEGUIR
 ▶▶▶-ESC-":SP=SP+2
4010 A$=INKEY$
4020 IF A$=CHR$(27)THEN RETURN
4030 GOTO 4010
5000 CIRCLE(X,Y),R:CIRCLE(X,Y),R-1
5010 FOR I=1 TO 6
5020 J=X+R*COS(A)
5030 K=Y+R*SIN(A)
5040 F%(I)=J:G%(I)=K
5050 A=A+B:Z=Z+1
5060 LINE(X,Y)-(F%(I),G%(I))
5090 IF I=6 THEN PRESET(126,40):PRINT#1,
"?"
5100 NEXT
5110 RETURN
6000 FOR I=1 TO 8
6010 READ M$(I)
6020 NEXT
6050 RETURN
7000 GOSUB 3000
7010 PRESET(170+EA,60)
7020 PRINT#1,M$(1)
7030 PRESET(170+EB,95)
7040 PRINT#1,M$(2)
7050 PRESET(126+EC,115+EF)
7060 PRINT#1,M$(3)
```

```
7070 PRESET(83+ED,95)
7080 PRINT#1,M$(4)
7090 PRESET(83+EE,60)
7100 PRINT#1,M$(5)
7110 RETURN
8000 COLOR 1:GOSUB 7000
8100 PRESET(157,160):PRINT#1,"SOLUCION =
 ?"
8105 PRESET(155,160)
8108 PRINT#1,"SOLUCION =  "
8110 PRESET(10,160):PRINT#1,"EXACTO";"="
;M$(6)
8120 PRESET(10,180):PRINT#1,"PARA SEGUIR
 ▶▶▶-ESC-"
8490 COLOR 10
8500 RETURN
8600 REM
8610 PRESET(10,160):COLOR 1
8620 PRINT#1,"NO ◆ REPITA"
8630 COLOR 10
8640 RETURN
8800 COLOR 1:PSET(5,10):PRINT#1,"+++++++
++"
8810 COLOR 1:PSET(5, 8):PRINT#1,"+++++++
++":RETURN
8900 COLOR 2
8910 PRESET(217,30):PRINT#1,"+++"
8920 REM PRESET(218,30):PRINT#1,"+++"
8930 COLOR 10:RETURN
9000 CLS:COLOR 15,4,4
```


GANE
7.000 PTAS.
todos los meses
PARTICIPANDO
EN NUESTRO
CONCURSO
MSX Magazine premiará cada mes los programas que nos hagan llegar nuestros lectores.
Para participar en este concurso abierto, todo aficionado a los ordenadores con este estándar deberá hacer llegar a la redacción de la revista el listado, un cassette y un texto explicativo.
Entre todos los programas que recibamos cada mes, serán seleccionados para su publicación aquellos que reúnan los siguientes criterios:
— Originalidad de la aplicación.
— Simplicidad del método de programación.
La única condición para participar en el concurso será que los programas no hayan sido publicados previamente en ninguna revista.
Enviar vuestros programas a: MSX Magazine
C/Bravo Murillo, 377 - 5.º A 28020 MADRID

```
MODELO CIRLET
   10 CLS
   20 DEFSNG A - Z: DIM F%(200),G%(200)
   30 INPUT "TOTAL DE ESPACIOS"; L
   40 INPUT "RADIO CENTRAL"; R
   50 X = 128: Y = 86
   60 A = 0: B = 2*3.14/L:W = 360/L
   70 OPEN "GRP:" AS#1
   80 SCREEN 2
   90 FOR I = 1 TO L
  100 J = X - R COS(A)
  110 K = Y - R SIN(A)
  120 F%(I)= J: G%(I) = K
  130 PRESET (J,K)
  140 PRINT #1, CHR$(Z+33)
  150 A =A+B: Z = Z+1
  160 NEXT
  170 GOTO 170
```

Figura 6

En la línea 70 se abre un archivo para inscripciones en 140, previa indicación del lugar en 120, de los símbolos deseados. En el modelo se emplean los signos por orden de caracteres (CHR$).

Problemas de lógica

Otra aplicación derivada de la materia precedente, mucho más amena es la solución de los círculos lógicos alfanuméricos que casi todos hemos intentado resolver alguna vez. Los hay verdaderamente complicados. Nuestro modelo es bastante sencillo, pero se puede agrandar a voluntad y por ello complicarlo convenientemente como ya expondremos oportunamente.

En primer lugar, y para no confundir las cadencias, empleamos el siguiente alfabeto:

A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

en el que se tienen presente las letras más frecuentes en los teclados MSX.

```
9002 IF SP=32 THEN EV$="EXCELENTE"
9005 IF SP>25 AND SP<32 THEN EV$="SOBRES
ALIENTE"
9010 IF SP>16 AND SP<25 THEN EV$="NOTABL
E"
9015 IF SP>8 AND SP<15 THEN EV$="BIEN"
9020 IF SP>4 AND SP<8 THEN EV$="REGULAR"
9025 IF SP<5 THEN EV$="INTENTELO DE NUEV
O"
9030 PRESET(10,10):PRINT#1,"
PUNTUACION"
9035 PRINT#1,"            ↑↑↑↑↑↑↑↑↑↑"
9040 PRINT#1,:PRINT#1,:PRINT#1,
9050 PRINT#1," HA OBTENIDO";SP;"PUNTOS"
9060 PRINT#1,:PRINT#1,
9070 PRINT#1,"          EVALUACION TOTAL="
:PRINT#1,:PRINT#1,: PRINT#1,"
     ";EV$
9100 PRESET(10,170):PRINT#1,"SEGUIR S/N"
9110 F$=INKEY$
9120 IF F$="S" OR F$="s" THEN RUN
9130 IF F$="N" OR F$="n" THEN END
9140 GOTO 9110
```

ANTES DE INTRODUCIRLO EN EL MSX LEED LAS ACLARACIONES POSTERIORES AL LISTADO, YA QUE LA IMPRESORA EN LA QUE LO HEMOS LISTADO DESPUES DE COMPRÓBAR SU CORRECTO FUNCIONAMIENTO, NO TIENE LOS SIGNOS IGUALES A LOS DE MSX.

Conviene recordar brevemente que las cadenas van encerradas entre comillas. Así:

"Gracias" "Diez" "De Pedro"
"Izquierdo" "Centeno"

son eslabones de una cadena suma de todos ellos, en este caso de letras, pero también podían ser números y números/letras.

Cada DATA contiene entre comas (,) cadenas, datos numéricos o datos alfanuméricos pero sin valor matemático real.

En las líneas 985, 986, 8800, 8810, 8910, 8920 sustituir los signos + por un ■ o algo similar.

En 9030 antes de PUNTUACION situar 12 espacios en blanco y en la 9070 antes de EVALUACION ocho espacios.

Pueden variarse las líneas de espacios con:

Space$(12) SPC(12)
STRING(12, "▲") siendo ▲
espacio en blanco.

La 9035 en lugar de flechas con dirección vertical alta emplear:

GRPH + U

Este modelo consiste en la prueba de un test sencillo para responder a la casilla número seis de 16 pruebas o elementos del total.

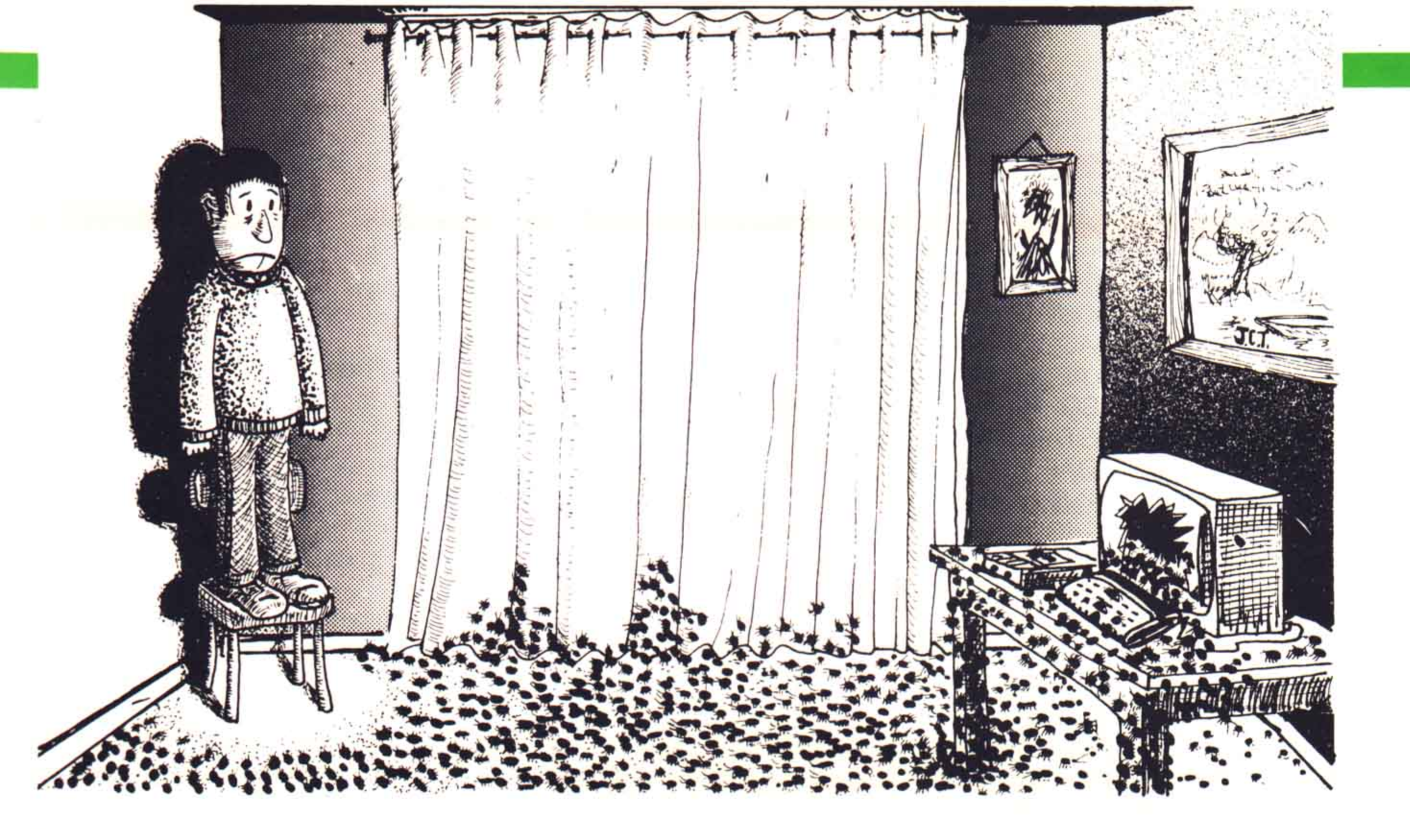
Hay que contestar con el número de caracteres justo y en el caso de decimales basta con dos. Se cierran paréntesis y se indican si los hay, es decir, todo tipo de signos, letras y números que se precisen para responder adecuadamente.

También existe un control de tiempo parcial y total y una evaluación final que no pretende ser exaustiva. Ni se crea superior el que más nota saque ni inferior el que menos. Es un juego pero sin extraterrestres.

Por la sencillez del modelo no necesita mayor comentario.

José Leal Rodríguez.





Ecología de campo

La palabra «ECOLOGIA» está, hoy en día, en boca de todo el mundo gracias al interés concedido por los medios de comunicación a los asuntos relativos al medio ambiente.

La ecología estudia las relaciones de los organismos con sus ambientes naturales. Esto no se consigue fácilmente, todo lo contrario, es un trabajo arduo de investigación de equipso de científicos y cada equipo dividido en su especialidad, como la ecología microbiana, vegetal, marina, de micromamíferos, etc. Una de las tareas más importantes en estas investigaciones es la recolección de muestras y datos que posteriormente son analizados muchas veces con ayuda de un ordenador debido a la gran cantidad de datos que se obtienen. También nosotros, con nuestro MSX, podemos obtener resultados interesantes. Analizando los datos nos aparecerán varias incógnitas o preguntas, como por ejemplo: ¿Por qué en dos sitios aparentemente similares la vegetación es distinta? y muchas más y son estas preguntas que distinguen un científico de otra persona y la diferencia entre un investigador bueno o malo es que el primero suele encontrar la respuesta. La ecología de campo es una de las pocas ramas donde un principiante puede llegar a conclusiones interesantes manteniendo un mínimo de rigor científico y descubrir algo nuevo, lo único que se necesita es paciencia y constancia, lo demás lo conseguiremos con la práctica y experiencia.

Ahora comentaré las distintas partes del programa y su utilidad.

Inventario y censo. Tal como indica consiste en hacer un inventario del número de organismos que nos encontramos en una zona delimitada y prefijada. Conviene, sobre todo al principio, estudiar zonas pequeñas y paulatinamente podemos aumentar la zona. Cada inventario se referirá a los animales de un mismo filum o tipo y con llegar a averiguar el orden o la familia tendremos suficiente (en un principio). Una vez familiarizados con las claves de identificación podemos llegar a género o especie. Es necesario hacer llegar a resultados interesantes. Es importante apuntar la mayor cantidad de datos posibles de la zona a estudiar. Cuanto más datos mejor, aunque os veáis en la obligación de ampliar el programa.

Respecto a la sociabilidad he distinguido cuatro grados pero siempre podéis cambiarlo si lo veis necesario.

1. Especies aisladas.
2. Especies en grupos pequeños.
3. Especies en grupos grandes.
4. Poblaciones.

Afinidad. Consiste en averiguar la relación que puede haber entre dos inventarios o entre dos especies. La afinidad entre inventarios

viene expresada por el coeficiente de Jaccard cuya fórmula es la siguiente:

$$\frac{C}{A+B-C}$$

La afinidad entre especies viene expresada por X^2 y no tengo más remedio que aconsejaros algún libro de bioestadística si necesitáis una explicación.

Estimación población total. Para hallar la población total he utilizado el índice de Lincoln por ser entre el más sencillo.

$$\text{Indice de Lincoln} = \frac{X1 \star X2}{Y}$$

Notas acerca del programa. Las distintas partes del programa están bien delimitadas y sin ninguna dificultad pueden ser alteradas según las necesidades de cada uno. La opción «Histograma» sólo tiene salida por impresora ya que sólo lo utilizo como documentación. La opción «Estimación población total» sólo tiene salida por pantalla ya que este resultado suele utilizarse sólo para incluirlo en algún texto o artículo y para inventarios de población total. Todas las demás opciones tienen salida tanto por pantalla como impresora. Las opciones de grabar y cargar no soy muy necesarias pero las he añadido para aquellos usuarios que no tienen impresora y quieren mantener los datos almacenados.

Juan Manuel Rial

```
10 SCREEN 0:KEYOFF:X=0:COLOR 3,1
20 CLS:PRINTSPC(6)"ECOLOGIA DE C
    AMPO":PRINT
30 PRINTSPC(8)"1-Inventario."
40 PRINTSPC(8)"2-Afinidad."
50 PRINTSPC(8)"3-Histograma."
60 PRINTSPC(8)"4-Estimación pobl
    ación"
70 PRINTSPC(10)"total."
80 PRINTSPC(8)"5-Grabar."
90 PRINTSPC(8)"6-Cargar."
100 PRINTSPC(8)"7-Introducción d
    atos"
110 Z$=INPUT$(1):ON VAL(Z$)GOSUB
    240,360,870,780,1100,1200,1
    30
120 GOTO 20
130 '«« INTRODUCCION DE DATOS »»
140 CLS:CLEAR 5000:INPUT"Nº de e
    species a tratar(max 50)";N
    :IF N>50 THEN 140
150 DIM A$(N),I%(N),P%(N)
160 CLS:INPUT"Nº de inventario";
    V%:INPUT"Hora(HH:MM)";R$
170 INPUT"Fecha(DD/MM/AA)";F$:IN
    PUT"Tipo de habitat";H$:
180 INPUT"Localidad";L$:INPUT"Al
    titud(m)";E%
190 INPUT"Sup.considerada(m²)";S
    %:INPUT"Temperatura(Cent.)"
    ;T%
200 CLS:FOR M=1 TO N:PRINT" ";M;
    " ";:INPUT"Especie";A$(M)
210 INPUT"Nº de individuos";I%(M
    ):INPUT"Sociabilidad(1-4)";
    P%(M):X=X+I%(M):CLS:NEXT M
220 GOTO 20
230 ' «« INVENTARIO »»
240 CLS:Y=0:GOSUB 1350:INPUT"Pan
    talla(1),impresora(2) o men
    u(3)";Y
250 ON Y GOSUB270,990,20
260 GOTO 240
270 CLS:TIT$="Especie
        abun      socia.":PRINTT
    IT$
280 FOR M=1 TO N:IF LEN(A$(M))>2
    0 THEN PRINT TAB(0)LEFT$(A$
    (M),20);ELSE PRINT TAB(0)A$
    (M);
290 AB!=(100*I%(M)/X):PRINT TAB(
    20)AB!;"%";TAB(32)P%(M)
300 IF N=21 AND M=21 OR N=42 AND
     M=42 THEN 320
310 IF VPEEK(882)<>32 THEN 340 E
    LSE NEXT
320 Z$=INPUT$(1):IF Z$<>" "THEN
    320
330 RETURN
340 Z$=INKEY$:IF Z$<>" "THEN 340
350 CLS:PRINTTIT$:NEXT
360 '«« AFINIDAD »»
370 CLS:LOCATE 4,4:PRINT"AFINIDA
    D"
380 LOCATE 4,6:PRINT"1-Entre inv
    entarios"
390 LOCATE 4,8:PRINT"2-Entre esp
    ecies"
400 LOCATE 4,10:PRINT"3-Menu"
```

```
410 Z$=INPUT$(1):ON VAL(Z$)GOSUB
      430,490,20
420 GOTO 370
430 CLS:A=0:B=0:C=0:J=0
440 INPUT"Nº de especies en inve
     ntario a";A:INPUT"Nº de esp
     ecies en inventario b";B:IN
     PUT"Nº de especies comunes
     a los dos";C
450 J=C/(A+B-C):PRINT:PRINT"Coef
     . de Jaccard=";J
460 LOCATE 0,20:PRINT"Menu=espac
     io,Impresora=tecla"
470 Z$=INPUT$(1):IF Z$<>" "THEN
     GOSUB620
480 RETURN
490 CLS:A=0:B=0:C=0:D=0:NM=0
500 INPUT"Nombre especie a";E1$:
     INPUT"Nombre especie b";E2$
510 INPUT"Nºde inventarios con e
     specie a y b";A
520 INPUT"Nºde inventarios con b
      y sin a";B
530 INPUT"Nºde inventarios con a
      y sin b";C
540 INPUT"Nºde inventarios sin a
      ni b";D
550 NM=A+B+C+D
560 JI=(NM+(B*C-A*D)^2)/((A+B)*(
     C+D)*(A+C)*(B+D))
570 PRINT:PRINT"JI²=";JI
580 LOCATE 0,20:PRINT"Menu=espac
     io,Impresora=tecla"
590 Z$=INPUT$(1):IF Z$<>" "THEN
     GOSUB660
600 RETURN
610 '«« IMPRESORA »»
620 CLS:GOSUB 750:PRINT"IMPRIMIE
     NDO"
630 LPRINT"Nº de especies en inv
     entario a=";A:LPRINT"Nº de
     especies en inventario b=";
     B:LPRINT"Nº de especies com
     unes a los dos=";C
640 LPRINT:LPRINTSPC(8)"Coeficie
     nte de Jaccard=";J:LPRINT:L
     PRINT
650 RETURN
660 CLS:GOSUB 750:PRINT"INPRIMIE
     NDO"
670 LPRINT"Tabla de contingencia
     ."
680 LPRINT:LPRINT:LPRINTSPC(26)"
     Especie a":LPRINTSPC(23)"Pr
     esente  Ausente":
690 LPRINT"Especie b   Presente"
     ;TAB(25) A;TAB(35)B;TAB(42)
     A+B
700 LPRINTSPC(12)"Ausente";TAB(2
     5)C;TAB(35)D;TAB(42)C+D
710 LPRINTTAB(25)A+C;TAB(35)B+D;
     TAB(42)NM:LPRINT
720 LPRINT"Especie a :";E1$,"Esp
     ecie b :";E2$
730 LPRINT:LPRINT"JI²=";JI
740 RETURN
750 CLS:PRINT"TIENE CONECTADA LA
      IMPRESORA(S/N)":
760 Z$=INPUT$(1):IF Z$="N" OR Z$
     ="n" THEN 20 ELSE RETURN
770 ' «« ESTIMACION »»
780 CLS:X1=0:X2=0:Y=0
790 PRINT"Estimación población t
     otal.":PRINT
800 INPUT"Nº de animales captura
     dos , marca-   dos y libera
     dos en el primer mues-   tr
     eo ";X1
810 INPUT"Nº de capturados en el
      segundo muestreo ";X2
820 INPUT"Nº de individuos del s
     egundo muestreoque ya habia
     n sido marcados en el    pr
     imer muestreo";Y
830 PRINT:PRINT"Indice de Lincol
     n=";:PRINT (X1*X2)/Y
840 Z$=INPUT$(1):IF Z$<>" "THEN
     GOSUB 840
850 RETURN
860 '«« HISTOGRAMA »»
870 CLS:GOSUB 1350:GOSUB 750:PRI
     NT"INPRIMIENDO"
880 LPRINTCHR$(14)"Histograma ce
     nso inventario nº";V%:LPRIN
     T:LPRINTCHR$(15)
890 LPRINT"  Especie
                          Nº indivi
     duos"
900 LPRINTTAB(40)"0";TAB(50)INT(
     X/4);TAB(60)INT(X/2):LPRINT
     TAB(50)"|"; TAB(60)"|"
```

```
910 LPRINTTAB(40)"————————————————————————————————————————"
920 FOR M=1 TO N:LPRINTA$(M);:Q=(40-LEN(A$(M))):FOR D=1 TO Q:LPRINT".";:NEXT D
930 IF I%(M)<X/40 THEN 960
940 O=(40*I%(M))/X:FOR D=1 TO O:LPRINT"█";:NEXT D:LPRINT I%(M):NEXT M
950 RETURN
960 IF I%(M)<(X/40)/3 THEN LPRINT"▌";I%(M) ELSE IF I%<(X/40)/2 THEN LPRINT"▌";I%(M) ELSE LPRINT "█";I%(M)
970 NEXT M
980 RETURN
990 ' «« INVENTARIO IMPRESORA »»
1000 CLS:GOSUB 750:PRINT"IMPRIMIENDO"
1010 LPRINTCHR$(14)"Inventario nº ";V%:LPRINTCHR$(15)
1020 LPRINT"FECHA:";F$;TAB(20)"HORA:";R$;TAB(40)"TIPO DE HABITAT:";H$
1030 LPRINT"LOCALIDAD:";L$;TAB(40)"SUPERFICIE CONSIDERADA:";S%;" m²"
1040 LPRINT"TEMPERATURA:";T%;" ºC";TAB(40)"ALTITUD:";E%;"m"
1050 LPRINT:LPRINT"Especie"TAB(22)"Abundancia(%)"TAB(40)"Sociabilidad":FOR M=1 TO 52:LPRINT"-";:NEXT M:LPRINT
1060 FOR M=1 TO N:IF LEN(A$(M))>22 THEN LPRINT LEFT$(A$(M),22);ELSE LPRINT A$(M);
1070 AB!=(100*I%(M)/X):LPRINT TAB(24)AB!;:LPRINTTAB(44)P%(M):NEXT M
1080 RETURN
1090 ' «« LECTO-GRABADORA »»
1100 CLS:PRINT"PULSE SAVE+LOAD DESPUES ESPACIO":Z$=INPUT$(1):IF Z$<>" "THEN 1100
1110 OPEN"CAS:TUL$"FOR OUTPUT AS #1
1120 PRINT#1,V%:PRINT#1,R$:PRINT#1,F$
1130 PRINT#1,H$:PRINT#1,E%:PRINT#1,S%
1140 PRINT#1,T%:PRINT#1,N:PRINT#1,X:PRINT#1,L$
1150 CLOSE#1
1160 OPEN"CAS:TUL$" FOR OUTPUT AS#1
1170 FOR M=1 TO N:PRINT#1,A$(M):PRINT#1,I%(M):PRINT#1,P%(M):NEXT M
1180 CLOSE #1
1190 RETURN
1200 CLS:PRINT"SI CARGA DÁTOS DESDE LA LECTO-GRABA- DORA PERDERA TODOS LOS DATOS QUE HAYAINTRODUCIDO"
1210 PRINT"QUIERE VOLVER AL MENU (S/N)":Z$=INPUT$(1):IF Z$="S"ORZ$="s" THEN 20
1220 PRINT"PULSE LOAD , DESPUES ESPACIO":Z$=INPUT$(1):IF Z$<>" "THEN 1220
1230 CLEAR 5000
1240 OPEN "CAS:TUL$"FOR INPUT AS #1
1250 INPUT#1,V%:INPUT#1,R$:INPUT#1,F$
1260 INPUT#1,H$:INPUT#1,E%:INPUT#1,S%
1270 INPUT#1,T%:INPUT#1,N:INPUT#1,X:INPUT#1,L$
1280 CLOSE#1
1290 DIM A$(N+1),I%(N+1),P%(N+1)
1300 OPEN "CAS:TUL$" FOR INPUT AS#1
1310 FOR M=1 TO N:INPUT#1,A$(M):INPUT#1,I%(M):INPUT#1,P%(M):NEXT M
1320 CLOSE#1
1330 GOTO 20
1340 ' ««RUT. COMPROBACION »»
1350 CLS:FOR M=1 TO N:IF A$(M)<>"" OR I%(M)<>0 OR P%(M)<>0 THEN NEXT M:RETURN
1360 PRINT"ERROR EN LOS DATOS.VUELVE A INTRODU- CIRLOS.NO DEJE ESPECIES SIN NOMBRE   NI INTRODUZCA NUMEROS = 0"
1370 PRINT:PRINT"PULSA ESPACIO"
1380 Z$=INPUT$(1):IF Z$<>" "THEN 1380
1390 RUN
```

Reloj digital



El programa crea un reloj digital que funciona mediante interrupciones, lo que hace que una vez activado se pueda seguir trabajando con el ordenador con plena libertad, sin que ello afecte al reloj.

El programa *BASIC* carga el código máquina, introduce los datos de horas, minutos y segundos y activa el parche de interrupciones (vector H-KEYI) que es llamado cincuenta veces por segundo por el sistema operativo.

La rutina activada utiliza un *«buffer»* creado entre &hdfe1 y &hdfea para actualizar los dígitos del reloj, y en la posición &hdfeb se mide cuándo ha transcurrido un segundo.

El resto del programa se encarga de actualizar los valores del *«buffer»*, mirar si se está en modo 1 de texto y si es así visualizar en pantalla los valores del *«buffer»*, que corresponde a los dígitos del reloj. Estos se visualizan en la parte de la pantalla que corresponde a la visualización de las teclas de función [F-5]-[F-10] a fin de no interferir en el resto de la pantalla.

El programa está pensado para funcionar con unidad de disco: a tal fin se quitará la sentencia CLS de la línea 30, se activará el NEW de la línea 140 y se grabará en el disco con el nombre «AUTOEXEC-BAS». Así, cuando conectemos el ordenador con el disco introducido se conseguirá una autoejecución con un maravilloso efecto.

Si no se utiliza la unidad de disco se puede utilizar el bloque de memoria entre &he061 y &hf37f para otras rutinas en código máquina.

La precisión del reloj está supeditada a la del generador de interrupciones del UDP, pero como no es normal que nuestro ordenador esté encendido durante un tiempo relativamente largo, nos será suficiente. Hay que tener en cuenta que mientras se utiliza el cassette para grabar o cargar programas nuestro rejoj permanecerá parado.

Juan Carlos Pérez Baza
León

```
1 '    Reloj digital permanente MSX.
2 '    Juan-Carlos Pérez Baza. 1986.
3 '
9 '      Inicialización y carga
              de código máquina
```

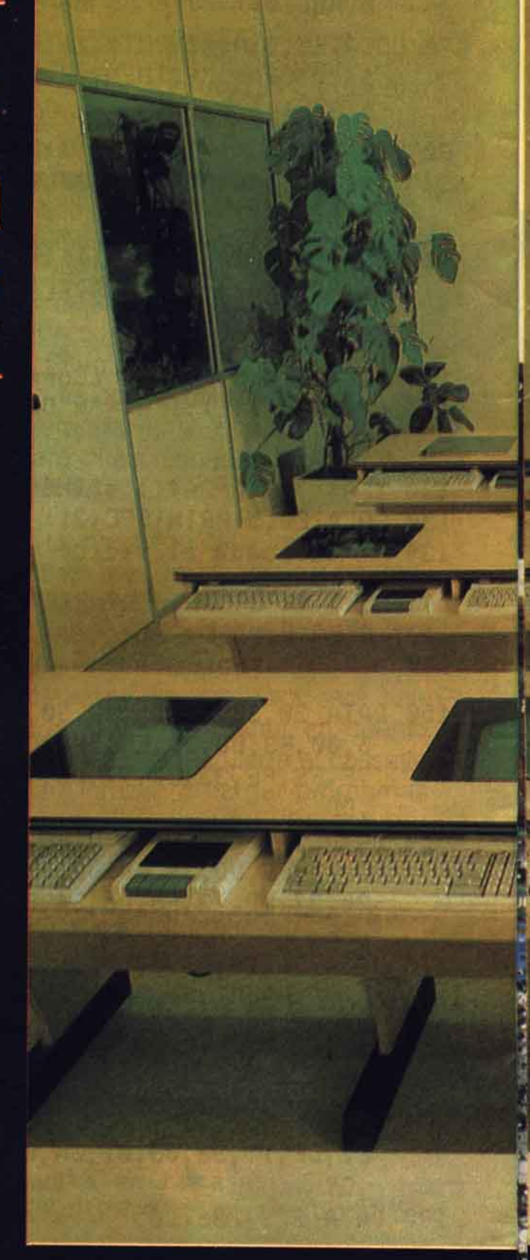
```
10 CLEAR200,&HDFE0
20 FORI=&HDFE1TO&HE060:READA$:A=VAL("
     &H"+A$):POKEI,A:NEXT
29 '         Puesta en hora
30 CLS:LOCATE0,5:PRINT"(HH:MM:SS)?"
```

PREMIO DE 7000 PTS

```
40 LOCATE1,5:B$=INPUT$(1):PRINTB$:IFB
     $<"0"ORB$>"2"THEN40ELSEPOKE&HDFE
     2,ASC(B$)
50 LOCATE2,5:A$=INPUT$(1):PRINTA$:IFB
     $<"2"AND(A$<"0"ORA$>"9")ORB$="2"
     ANDA$>"3"THEN50ELSEPOKE&HDFE3,AS
     C(A$)
60 LOCATE4,5:A$=INPUT$(1):PRINTA$:IFA
     $<"0"ORA$>"5"THEN60ELSEPOKE&HDFE
     5,ASC(A$)
70 LOCATE5,5:A$=INPUT$(1):PRINTA$:IFA
     $<"0"ORA$>"9"THEN70ELSEPOKE&HDFE
     6,ASC(A$)
80 LOCATE7,5:A$=INPUT$(1):PRINTA$:IFA
     $<"0"ORA$>"5"THEN80ELSEPOKE&HDFE
     8,ASC(A$)
90 LOCATE8,5:A$=INPUT$(1):PRINTA$:IFA
     $<"0"ORA$>"9"THEN90ELSEPOKE&HDFE
     9,ASC(A$)
100 LOCATE10,5:PRINT" ¿Correcto?"
110 A$=INPUT$(1):IFA$="N"ORA$="n"THEN
     30
120 IFA$<>"S"ANDA$<>"s"THEN110
130 LOCATE0,5:PRINTSPC(21)
139 '    Llamada al vector H-KEYI
140 FORJ=&HFD9CTO&HFD9ASTEP-1:READB:P
     OKEJ,B:NEXT:'NEW
149 '         Datos
150 DATA 20,30,30,3a,30,30,3a,30,30,2
     0,00,F3,DD,21,ED,DF,DD,34,FE,DD,
     7E,FE,FE,33,C0,97,DD,77,FE,1E,30
     ,DD
160 DATA 34,FC,DD,7E,FC,FE,3A,20,44,D
     D,73,FC,DD,34,FB,DD,7E,FB,FE,36,
     20,37,DD,73,FB,DD,34,F9,DD,7E,F9
     ,FE
170 DATA 3A,20,2A,DD,73,F9,DD,34,F8,D
     D,7E,F8,FE,36,20,1D,DD,73,F8,DD,
     34,F6,DD,7E,F6,FE,3A,20,10,DD,73
     ,F6
180 DATA DD,34,F5,DD,7E,F5,FE,36,20,0
     3,DD,73,F5,3A,AF,FC,FE,00,C0,21,
     E1,DF,11,B6,03,01,0A,00,CD,5C,00
     ,C9
190 DATA 223,236,205
```

SVI

El cassette y salida directa al altavoz



El intercambio de información entre nuestro ordenador y el cassette se produce a través del PPI o Interface Programable de Periféricos.

Como podemos ver en la figura 1 del número de diciembre, el PPI está compuesto de cuatro *ports* de los que dos son de entrada y dos son de salida.

Los *ports* 96 hex, 98 hex y 99 hex del PPI ya los hemos usado en el capítulo anterior para hacer una lectura directa del teclado y de los disparadores de los *joysticks*.

Los *ports* 96 hex y 97 hex los podemos usar también para producir sonidos sin la intervención del PSG, ya que comunican directamente con el altavoz. Utilizando el *port* 96 hex lograríamos el sonido mediante el envío alterno a dicho *port* de los números 128 y 0, o sea, alzando y bajando el *bit* 7 de dicho *port;* prueba lo siguiente:

```
10 OUT&H96,128
20 OUT&96,0
30 GOTO10
```

Si utilizamos el *port* 97 hex los números a alternar serían el 14 y el 15.

```
10 out&97,14
20 out&97,15
30 goto10
```

pero si utilizas este *port* cuidado con alzar el bit 7 del mismo, ya que el crack del ordenador sería inmediato; prueba lo siguiente para dejar «colgado» el ordenador.

```
OUT &h97,128
```

El manejo del cassette a través del PPI se consigue de la siguiente manera:

— A través del port 96 hex podemos encender y apagar el motor del cassette, el motor se parará

318/328



cuando alcemos el *bit* 4 (Recuerda que los *bits* del octeto se empiezan a contar desde cero). Para comprobarlo prueba alternativamente.

```
OUT&H96,16
OUT&H96,0
```

— Con el port 97 hex también podemos manejar el motor, ahora el motor se parará cuando enviemos un 9 al *port* y arrancará enviando un 8. Además este *port* nos permite grabar impulsos directamente en la cinta previa pulsación de las teclas *RECORD* y *PLAY* del *cassette* y enviando alternativamente los números 11 y 10 en espacios de tiempo precisos.

— La señal proveniente del *cassette* podemos leerla en el *port* 98 hex. El *bit* 6 alzado nos indica que el motor del cassette está parado, y un 0 en el mismo nos marca que el motor funciona. En el *bit* 7 tenemos presentes los impulsos provenientes de la cabeza lectora de forma que un 0 en este *bit* indica la ausencia de señal. Realizaremos a continuación dos programas que aplicarán algunas de las características aprendidas. El primero lee el *port* 98 hex, que recoge la señal del *cassette*, e imprime su contenido en binario en la pantalla a la vez que lo envía directamente al altavoz.

```
10 A=INP(&H98):
   A$=STRING$(8−LEN
   (BIN$(A)), "0")+BIN$(A)
20 OUT&H96,A AND &HEF
```

```
30 PRINT A$:GOTO 10
```

Ejecuta este programa en tu ordenador y a continuación haz que el *cassette* lea alguna cinta, bien con *software* o con música. Se puede comprobar que el parecido entre lo que se oye por el altavoz y lo que se supone debe haber en la cinta es muy relativo o nulo. Esto se debe naturalmente a la lentitud del *basic* para ejecutar instrucciones respecto a la velocidad de lectura de datos en el *cassette* y emisión por el altavoz.

Probaremos ahora el siguiente pulgoprograma en lenguaje de máquina.

```
PRINCI IN A,98       ; Información
                       del port 98
                       hex
       AND EF        ; Para no parar
                       el motor del
                       cassette
       OUT 96,A      ; Dato al altavoz
       CALL 3512     ; Prueba si
                       CTRL-STOP
       JRNC PRINCI; SI=>VUELVE
       RET           ; NO=>SALE
                       AL BASIC
```

Cargándolo mediante el siguiente programa *basic*.

```
10 CLEAR 100,&HD54E
20 FOR A=&HD54E TO
   &HD559
30 READ A$:POKEA,
   VAL("&H"+A$:NEXT
40 REFUSR=&HD54E
50 Z=USR(0)
60 DATA DB,98,E6,EF,D3,96,
   CD,12,35,30,F5,C9
```

Ahora no aparece información en la pantalla pero si ponemos una cinta de música en el *cassette* lo podremos oír en el altavoz de nuestro televisor o amplificador conectado.

Hemos logrado hacer pasar el sonido a través de nuestro ordenador, lo que demuestra que puede ser digitalizado con bastante calidad a pesar de haber contado con sólo dos niveles de volumen (los estados 1 y 0 del bit del *port* de acceso), se podrán obtener mejores efectos usando los 15 niveles de volumen del generador de sonido. Aquí empieza el trabajo de los sintetistas de voces y otros ruidos raros.

La impresora

Los *ports* 10, 11 y 12 hex nos comunican con el *interface* de datos en paralelo, que nos sirve de intermediario para manejar la impresora de un modo directo mediante las instrucciones «IN» y «OUT» tanto desde el *basic,* como desde el código máquina.

Como se indica en la tabla del número de diciembre, los *ports* 10 hex y 11 hex son de salida, para enviar mensajes a la impresora, mientras que el *port* 12 hex es de entrada y nos proporciona una señal proveniente de la misma que nos indica si está o no preparada para recibir datos.

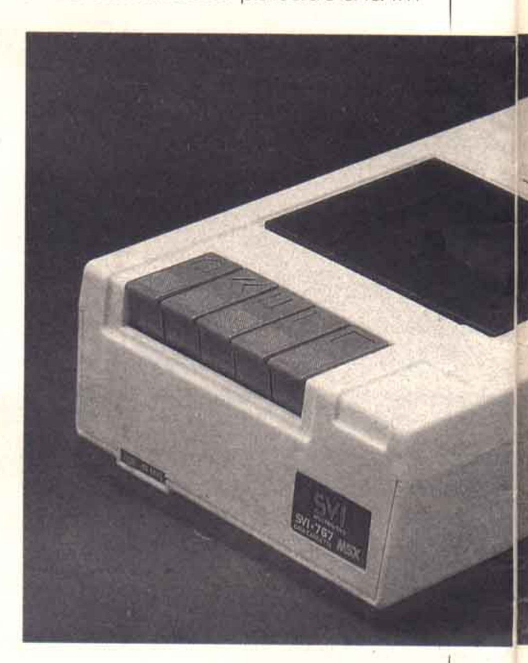
Cuando la impresora está fuera de línea, o sea, no preparada, se pone a 1 el *bit* 0 de este último *port,* y viceversa, cuando en el *bit* 0 hay un 0, sabemos que la impresora está dispuesta a aceptar nuestros datos y caracteres de control.

Para enviar la información a la impresora hemos de seguir el «protocolo» propio del *interface* paralelo y que explicamos a continuación.

En el *port* 10 hex colocamos *(OUT)* la información a enviar, ya sean datos o caracteres de control para la impresora (caracteres cuyos códigos ASCII abarcan de 0 a 31).

A continuación ponemos a 0 el *bit* 0 del *port* 11 hex, este *bit* se llama de sincronismo *(DATA STROBE),* e inmediatamente lo volvemos a poner a 1, con lo que la impresora acepta el dato colocado en el *port* de datos 10 hex. Este dato no se imprime inmediatamente, salvo en el caso de que la impresora no tenga memoria intermedia, sino que se almacena en ésta y sale al papel, bien cuando dicha memoria se llena, o cuando se recibe el carácter de control «13» (código de *ENTER*) o, en algunas impresoras, cuando al desconectarlas queda algún dato en dicha memoria o *«BUFFER».*

El siguiente ejemplo efectúa el envío del carácter pulsado a la impresora, a la vez que lo imprime en la pantalla.




```
10 A$=INKEY$:IF A$=" "
   THEN 10
20 PRINT A$;OUT&H10,ASC
   (A$):OUT&H11,0:OUT&11,1
30 GOTO10
```

Rutinas de la ROM y variables del sistema

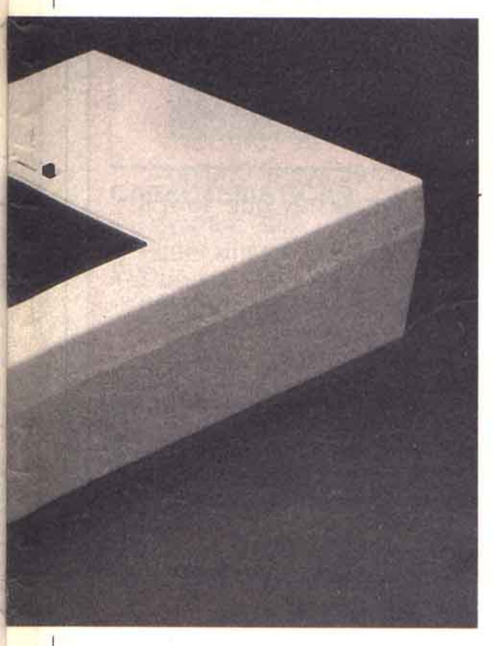
Aunque aún nos falta por des-

cubrir cómo funcionan en modo directo algunos periféricos tales como el disco, el modem o la tarjeta de 80 columnas, vamos a empezar a ver algunas rutinas de la memoria ROM y variables del sistema que nos ayudarán a conocer mejor nuestro ordenador.

Para ello nos ayudaremos de las tablas que insertamos en el número de diciembre, que muestran una lista de rutinas y variables con la dirección en que se encuentran en la memoria del ordenador.

— Los saltos de enganche:

Empezaremos por fijarnos que



las variables del sistema a partir de la dirección FE79 hex tienen delante la letra «H» y un punto. Esto indica que cada una de ellas es un *«HOOK JUMP»* o salto de enganche, a los que ya hicimos alusión en el primer artículo cuando presentamos un programa que nos conmutaba con el otro banco de memoria disponible en los SVI-328.

Una variable de estas características consiste en tres octetos de memoria, el primero de los cuales suele contener normalmente el código «C9» que pertenece a la instrucción en lenguaje ensamblador *«RET»* equivalente a la instrucción *basic «RETURN»*.

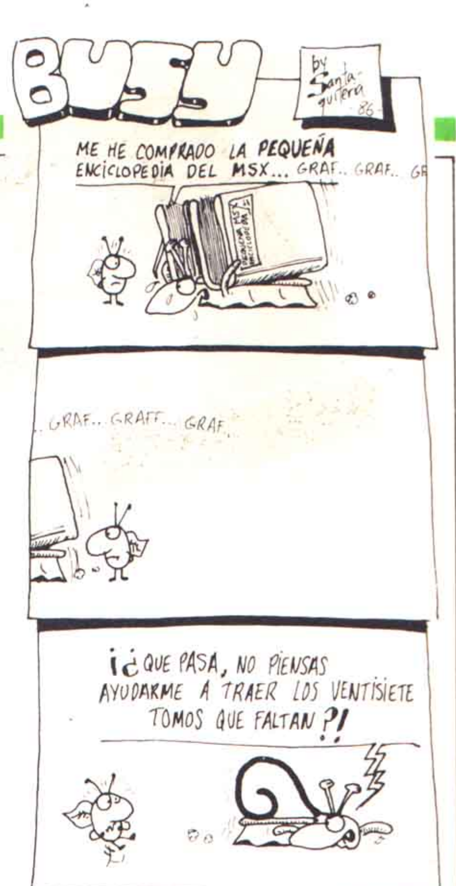
Hay algunas rutinas de la ROM que en determinado momento hacen una llamada *«CALL»* a estas zonas de memoria. Si el código encontrado pertenece a la instrucción *«RET»* el flujo del programa continúa en la instrucción siguiente a *«CALL»* en la rutina que se estaba efectuando.

Si se cambia el contenido de la variable, de modo que el primer octeto de los tres contenga un código de salto «JP» y los otros dos contengan la dirección a la que saltar, por ejemplo la dirección en que comienza nuestra rutina en código máquina, habremos conseguido desviar el flujo normal del programa en esa rutina ROM, y de esta forma habremos «enganchado» la rutina ROM con la nuestra pudiendo así definir un nuevo comando, o hacer que el ordenador se comporte de forma diferente a la habitual.

Dentro de los saltos hay algunos que están marcados con las letras S.D., lo cual indica que son usados por rutinas que pertenecen al sistema de disco.

Una de estas variables o saltos tiene mayor interés que todos los demás ya que es muy frecuentado, se trata del *H.GONE* que se encuentra en la dirección FF57 hex.

Este enganche es consultado por todas las declaraciones *BASIC* antes de la prueba de sintaxis. Esto significa que si hacemos un desvío en este punto, podemos construir una rutina que utilice nuestros propios comandos, pero para usarlo hemos de tener en cuenta unas cuantas reglas básicas:



1. Cuando se realiza la llamada el registro «A» contiene el código de la palabra a usar. Si la palabra es un comando o función *basic* el registro «A» contiene el código de dicho comando o función según las tablas de las figuras 2 y 3 que aparecieron en el número 13 de la revista. Nuestra primera instrucción en la rutina a la que se ha desviado el flujo ha de ser una comprobación de que el registro A contiene el código esperado, y si no es habrá un retorno a la rutina ROM inicial.

2. Tras realizar nuestra rutina la función para la que fue programada, una instrucción *RET* la enviaría de nuevo a la *ROM,* cuando normalmente querremos retornar al *basic.* Para conseguir esto hemos de retirar la dirección de retorno a la *ROM* de la pila, mediante un incremento doble del puntero de la misma, de modo que apunte a la dirección de retorno al *basic.*

3. En el registro HL se almacena la dirección de la posición,



dentro del programa *basic*, que se está ejecutando. Antes de retornar nuestra rutina debemos incrementar esta dirección de modo que apunte a la siguiente instrucción *basic*, y no a la que acabamos de analizar.

En el ejemplo siguiente creamos el comando «MUS» que hará que suene una melodía cada vez que lo escribamos.

Primero prueba el programa y luego analizamos el ensamblador fuente:

```
10 CLEAR 100,&hD000
20 FOR A=&hD000 TO
   &hD028
30 READ A$:POKE A,VAL ("&h"
   +A"):NEXT
40 POKE&HFF58,0:POKE
   &HFF59,5HFD0:POKE
   &HFF57,&HC3
60 DATA FE,4D,CO,23,7E,2B,
   FE,55,CO,23,23,7E
70 DATA 2B,2B,FE,53,CO,23,
   23,23,33,33,E5,21
80 DATA 1F,DO,CD,24,2C,E1,
   C9,22,42,43,44,45,46
90 DATA 46,41,43,0
```

Un detalle a tener en cuenta en la línea 40 es el orden en que se han introducido los datos para el enganche. Al estar introduciendo estos datos directamente desde el *basic*, hemos de tener cuidado y dejar como último dato a introducir el código de salto, pues cualquier otro orden provocaría un salto prematuro y un fallo en el programa cargador.

Tras teclear *«RUN»* y pulsar *«ENTER»* el ordenador devuelve el mensaje OK, con lo que tendremos los mismos comandos y posibilidades de antes (prueba *LIST*), pero tenemos un nuevo comando llamado «MUS» y ¿para qué sirve?, pruébalo y verás.

El ensamblador fuente es como sigue:

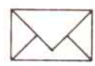
Dirección	Código		Comentarios
D00	------------CP 77	-----	"M"??
D002	------------RET NZ	-----	NO---> Retorna
D003	------------INC HL	-----	Siguiente carácter
D004	------------LD A, (HL)	-----	A recoge el código
D005	------------DEC HL	-----	Por si hay que volver.
D006	------------CP 85	-----	"U"??
D008	------------RET NZ	-----	
D009	------------INC HL	-----	
D00A	------------INC HL	-----	
D00B	------------LD A, (HL)	-----	
D00C	------------DEC HL	-----	
D00D	------------DEC HL	-----	
D00E	------------CP 83	-----	"S"??
D010	------------RET NZ	-----	
D011	------------INC HL	-----	
D012	------------INC HL	-----	
D013	------------INC HL	-----	Siguiente carácter basic.
D014	------------INC SP	-----	Elimina retorno
D015	------------INC SP	-----	a la ROM.
D016	------------PUSH HL	-----	Salva el puntero BASIC.
D017	------------LD HL, CMUS	-----	HL=Inicio cadena música
D01A	------------CALL 2C24	-----	Rutina "PLAY".
D01D	------------POP HL	-----	Recupera puntero BASIC
D01E	------------RET	-----	Final.
D01F	--CMUS--34,66,67,68,69,70,70,65,67,0		

A destacar del programa la llamada a la rutina *ROM "PLAY"* que interpreta la cadena de música a cuya dirección apunta el registro HL. Dicha cadena ha de empezar en comillas y terminar en cero.

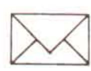
Venerando Solís

compro, vendo, cambio...

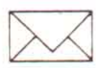
• Vendo ordenador Philips MSX-II, VG-8235. Prácticamente nuevo, con diskettes de programas, home-office, designer y MSX-DOS. Buen precio. Llamar a Raúl al Tel.: (93) 200 87 46.



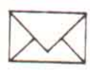
• Intercambio programas en cinta. Escribir a Juan Antonio López. Alvarado, 14. 08033 Barcelona. O llamar al Tel.: (93) 350 71 78.



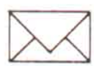
• Compro programas con relación a radio afición CW-RTTY, libro de guarda, etc. Escribir a Alejandro García. Avda. Hilario Ruiz, 46. 15621 Cabañas (La Coruña).



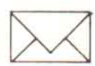
• Vendo por cambio a MSX II ordenador Spectravideo SVI-728, impecable por 25.000 ptas. Llamar a Rafael García al Tel.: (94) 493 30 58.



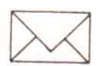
• Compro programas de comunicaciones entre ordenadores con modem. También deseo contactar con interesados en el tema, sobre todo aquellos que dispongan de modem. Escribir a Manuel López Cuesta. Puentelarra, 18. 28031 Madrid.



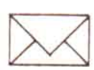
• Vendo ordenador Spectravideo SVI-328, Superexpander 605_B, dos unidades de disco DC/DD e impresora Seikosha, cable y tarjeta Centronics. Todo, en perfecto estado, por el 50% de su valor según factura de compra. Escribir a Jacinto Moreno Aguilar. Avda. de Cádiz, 10. 14009 Córdoba. O llamar al Tel.: 29 90 63.



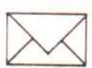
• Vendo ordenador SVI-728, nuevo, por 25.000 ptas. Escribir a Javier Pérez. Joan Bardina, 17. 08830 Sant Boi (Barcelona).



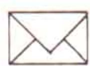
• Vendo juegos para MSX. Escribir a Luis Montero Campos. Extraradio, s/n. 16740 La Almarcha (Cuenca).



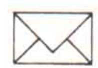
• Intercambio programas. Escribir a Juan Sánchez Sevilla. Batalla del Salado, 55. Tarifa (Cádiz). O llamar al Tel.: (965) 68 49 89.



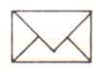
• Vendo grabadora Sanyo M-1110 con varios juegos, para comprar unidad de disco. Precio razonable. Interesados escribir a Santi Sales. Avda. Atarazanas, 11. 08001 Barcelona.



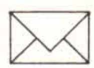
• Compro unidad de disco de 3.5 pulgadas para MSX. También intercambio programas de todo tipo. Escribir a Francisco Gimeno Carbonel. Ribera, 19. Silla (Valencia). O llamar al Tel.: 120 10 74.



• Vendo Philips VG-8010 con varios programas, interface Centronics, ampliación de 64K, cassette, cables y manuales, todo por 22.000 ptas. Llamar a Javier Torres al Tel.: (91) 204 43 15.

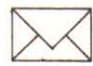


• Cambio juego para ordenador SVI-328. Interesados escribir a Jorge Carbonell Diaz. Paseo de la Chopera, 31. 28045 Madrid. O llamar al Tel.: (91) 473 56 57.



• Vendo diversos juegos. Llamar al Tel.: (987) 41 26 95, preguntar por Alberto. O al Tel.: (987) 41 05 30, preguntar por David.

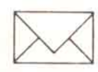




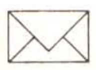
• Vendo Superexpander SVI-605A, monitor en fósforo verde, cartucho de 80 col., tableta gráfica y ordeandor SVI-328 con cassette. Regalo varios programas de aplicación. Interesados contactar con Venancio Gancedo. Avda. Valdecilla, 27. 39011 Santander (Cantabria). O llamar al Tel.: (942) 33 98 69.

compro, vendo, cambio...

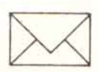
• Vendo ordenador Philips VG-8000, con manuales en castellano, cables de conexión, programas y varias revistas. Todo por 37.000 ptas. (negociables). Escribir a Fernando Alaban Albero. Guadalupe, 13. Villena (Alicante). O llamar al Tel.: (965) 80 38 33.



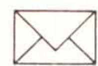
• VENDO videojuegos, regalo dos cartuchos a un precio interesante. Vendo Quickshot II o cambio por juegos. Estoy interesado en una ampliación de memoria de 64K. Escribir a Martín López Corredoira. Polígono del Ceao, 71. Lugo.



• Cambio programas para MSX-2, sólo en disco. Interesados escribir a Jordi Tuques Raimat, Alfred Perenya, 43. 25004 Lleida. O llamar al Tel.: (973) 23 52 61.

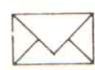


• Vendo ordenador SVI-328, cassettes SVI-904 y monitor fósforo verde. Todo ello en perfecto estado. Adjunto manual de uso, manual de referencia BASIC y numerosas revistas y programas. Todo por 30.000 ptas. Interesados contactar con Arturo Carnicer en el Tel.: (93) 317 15 22.

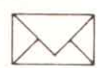


• VENDO urgente ordenador SVI-328 MKII, data cassette SVI-904, 5 juegos y más de 40 programas y joystick Quickshot II. Todo en perfecto estado y por 38.000 ptas. Llamar a Pedro Luis Gomis Faus al Tel.: (96) 241 43 64.

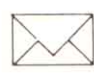
• CLUB MSX por carta y teléfono. Suministramos la más reciente información sobre programas y periféricos. ¡No pierdas la onda! Preguntar por Juan Luis al Tel.: (948) 23 63 85 o por Ignacio al Tel.: (948) 24 85 22. Llamar de 19 a 21h.



• INTERCAMBIO juegos para MSX, escribir a Juan Manuel Navarro Silva. Poeta Más y Ros. 46022 Valencia.



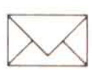
• INTERCAMBIO programas, juegos, gestión y aplicaciones. Escribir a Enrique Navarro. Primavera, 37, 2-1. Hospitalet (Barcelona).



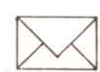
• Vendo programas en diskette de todo tipo. Contactar con Adolfo G. Barberá. Dr. Sanchís Bergón, 5-8. 46008 Valencia. Tel.: 331 26 57.



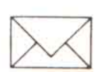
• Vendo ordenador MSX Hit-Bit 101P, con ampliación de memoria de 64K, por 27.000 ptas. Llamar a Antonio Montero Batle al Tel.: (93) 652 16 94.



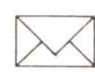
• VENDO SVI-328 (80K) con cassette, interface para impresora, dos joysticks, cartucho de juegos y programas diversos. Regalo libros y revistas sobre este ordenador. Todo por 35.000 ptas. Llamar a José María Sánchez Benito al Tel.: (985) 57 45 14 o escribir a: Sebastián Elcano, 10, 2-B. La Luz-Avilés, 33400 (Asturias).



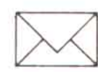
• CAMBIO Amstrad CPC-6128 con todo lo que trae y 2 discos con programas, por un SVI 738 X'press de MSX con pantalla o por un MSX de la II Generación. Escribir a Miguel Fernández. Plaza Gutiérrez. Semprún, 1, 1-A. 47012 Valladolid o llamar al Tel. 39 81 60.



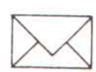
• Club LSD Printer. Todo en software y hardware para los usuarios del MSX. Intercambio programas. Apartado 2093. Murcia.



• Vendo Philips VG-8010 en perfectas condiciones, con sólo cinco meses. Adjunto manuales. Todo por 25.000 ptas. Escribir o llamar a Ramón Otero Pulin. Costaña, 8. 27500 Chantada (Lugo). Tel.: (982) 44 05 15.



• Intercambio programas. Contactar con H.M.P. Gral. Franco, 53. Los Cristianos II, 2.°-23. Los Cristianos (Tenerife).



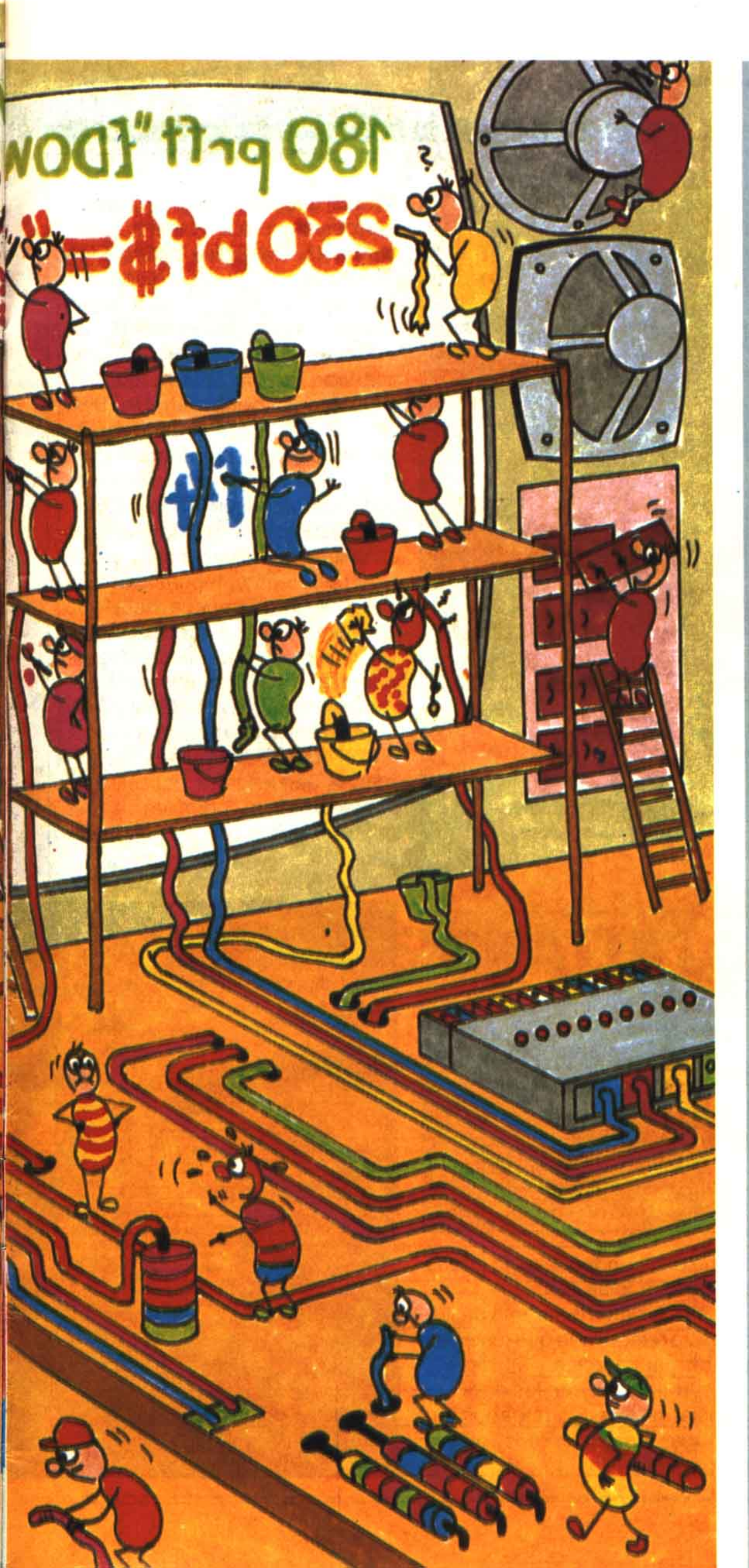
• Vendo programas por haber vendido el ordenador. Llamar a Juan al Tel.: (957) 27 11 80 (de 3 a 5).

Código máquina MSX

En esta ocasión vamos a abordar el problema de diseñar una rutina que nos puede resultar útil, especialmente para buscar *«pokes»* para juegos. Se trata de una rutina para buscar en la memoria las direcciones en que se encuentra una determinada secuencia de códigos. Así, si queremos averiguar dónde hay alguna llamada a la subrutina del sistema operativo encargada de imprimir un carácter (00A2H), la secuencia a buscar sería la formada por los códigos hexadecimales CD, A2 y 00.

Además, vamos a empezar a acostumbrarnos a realizar ordinogramas o diagramas de flujo. Estos son esos dibujitos formados por líneas, flechas, rombos, rectángulos, palabras, etc., que generalmente nadie entiende, pero que en realidad son bastante sencillos de descifrar, y muy útiles para desarrollar nuestras ideas y convertirlas en programas, no sólo en código máquina sino también en





cualquier otro lenguaje.

En esta ocasión sólo utilizaremos cuatro símbolos. El rectángulo normal indica una acción simple, que en código máquina se traducirá normalmente en una o unas pocas instrucciones. El rectángulo con lados redondeados indica una etiqueta de comienzo de programa o de fin de programa. El rectángulo con los lados verticales dobles representa una acción compleja compuesta por muchas instrucciones, y que generalmente se traduce como una llamada a subrutina (o, en PASCAL, un procedimiento). Por último, los rombos indican una toma de decisión, que puede seguir una u otra bifurcación según sea la respuesta a la pregunta indicada en su interior. En código máquina se traducen generalmente en una instrucción condicional.

Existen diversas formas de buscar una cadena en memoria, algunas muy rápidas en su ejecución, otras muy cortas. Nosotros hemos elegido una que sea comprensible fácilmente para los que siguen este curso.

De manera intuitiva podemos definir claramente el proceso que seguiremos para buscar una secuencia de códigos: buscamos el primero; cuando lo encontramos, miramos si a continuación se halla el segundo, y así hasta completar la secuencia. Si conseguimos completarla, imprimimos en pantalla la dirección del primer código, pero si nos falla alguno de los caracteres de la secuencia, retomamos la dirección del primer código, la aumentamos en uno y seguimos buscando.

El ordinograma adjunto refleja este mismo proceso con un análisis más detallado. Nuestra rutina nos va a permitir decidir a partir de qué dirección de memoria empezamos a buscar, y hasta cuál otra

COMIENZO

PUNTERO DE MEMORIA = COMIENZO

PRESERVA PUNTERO DE MEMORIA

PUNTERO DE CADENA = PRIMERO

CONTADOR = NUMERO DE CARACTERES

CARGA ACUMULADOR CON (PUNTERO DE CADENA)

COMPARA CON (PUNTERO DE MEMORIA)

INCREMENTA PUNTERO DE MEMORIA

IGUALES? — SI / NO

INCREMENTA PUNTERO DE CADENA

DECREMENTA CONTADOR

CONTADOR=Ø? — NO / SI

RECUPERA PUNTERO DE MEMORIA

IMPRIME PUNTERO DE MEMORIA

ESPERA PULSACION DE TECLA

"F" o "f" — NO / SI

RECUPERA PUNTERO DE MEMORIA

INCREMENTA PUNTERO DE MEMORIA

PUNTERO DE MEMORIA MENOR QUE LIMITE — SI / NO

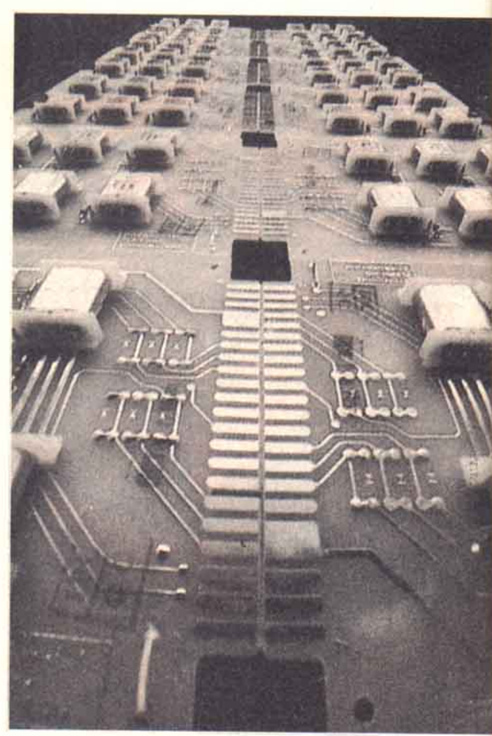
FIN

continuamos la búsqueda.

Podremos buscar cualquier secuencia formada por entre uno y 255 caracteres. Además, para que nos dé tiempo a apuntar las direcciones impresas en la pantalla, tras la rutina de impresión esperaremos a que se pulse una tecla. Por último, hemos previsto la posibilidad de abandonar la búsqueda pulsando la tecla «F».

Vamos a ver el diagrama de flujo paso a paso. Dado que tenemos que buscar caracteres en memoria y compararlos con los de la secuencia, también guardados en memoria, necesitaremos utilizar dos punteros. Uno de ellos apuntará al carácter de la secuencia que estamos buscando, y le llamaremos PUNTERO DE CADENA. El otro apuntará a la dirección de memoria por la que ve la búsqueda, y le llamaremos PUNTERO DE MEMORIA.

Como véis, lo primero que hacemos es inicializar el PUNTERO DE MEMORIA con la dirección en la que queremos comenzar la búsqueda. Esta dirección nos interesa guardarla en el *Stack* (con una instrucción *PUSH*), ya que, si localizamos la secuencia, es la que deberemos imprimir, pero el proceso de búsqueda hará que esté apuntando a la dirección siguiente al último carácter de la secuencia. A continuación inicializamos el PUNTERO DE CADENA, y además inicializamos un contador con el número de códigos que forman la secuencia a buscar. Este contador nos permitirá saber cuándo hemos completado la cadena.

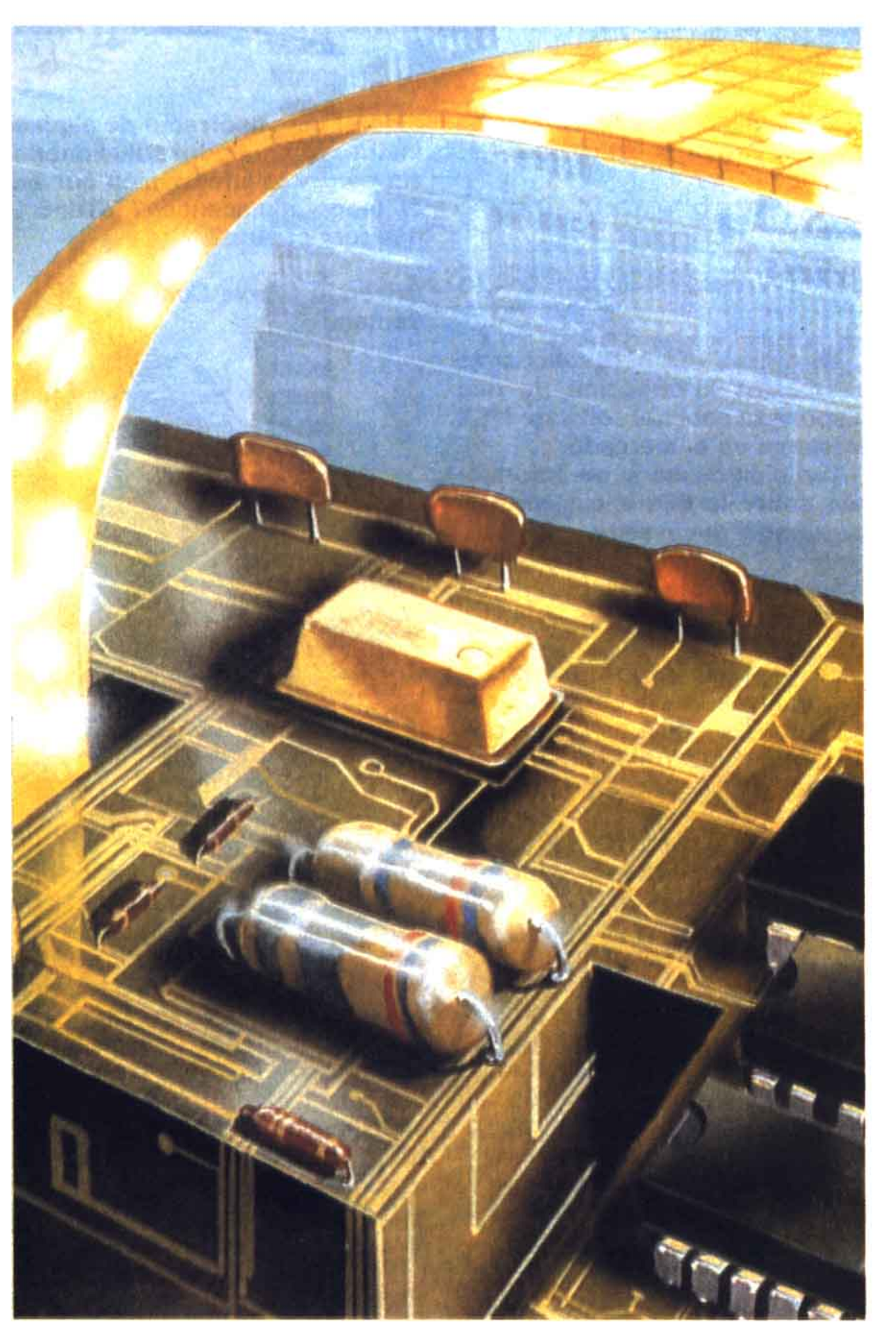


Ahora empezamos a buscar. Cargamos al acumulador con el código al que apunta el PUNTERO DE CADENA (primer código de la secuencia) y lo comparamos con el código al que apunta el PUNTERO DE MEMORIA. A continuación preparamos la siguiente búsqueda incrementando el PUNTERO DE MEMORIA y verificamos si ambos códigos eran iguales. Evidentemente, si no lo eran deberemos continuar con el siguiente carácter en memoria. Aquí observaréis que, de forma aparentemente absurda, recuperamos del *stack* el PUNTERO DE MEMORIA y lo incrementamos, cosa que ya habíamos hecho tras la comparación. Efectivamente, si la comparación



código máquina

```
10 WIDTH 40
20 CLEAR 200,&HEE00
30 GOSUB 200
40 INPUT"Longitud de la secuencia";L:IF
   L<0 OR L>255 THEN 40
50 INPUT"Direccion de comienzo";C$:IF LE
   N(C$)<>4 THEN 50
60 INPUT"Direcion final";F$:IF LEN(F$)<>
   4 THEN 60
70 POKE &HEE5F,VAL("&H"+RIGHT$(C$,2))
80 POKE &HEE60,VAL("&H"+LEFT$(C$,2))
90 POKE &HEE61,VAL("&H"+RIGHT$(F$,2))
100 POKE &HEE62,VAL("&H"+LEFT$(F$,2))
110 POKE &HEE63,L
120 FOR N=1 TO L
130 PRINT"Introduce el caracter";N;
140 INPUT" :";V
150 POKE &HEE63+N,V
160 NEXT
170 DEF USR=&HEE00
180 A=USR(0)
190 END
200 FOR N=0 TO &H5E
210 READ V$:V=VAL("&H"+V$)
220 POKE &HEE00+N,V
230 NEXT:RETURN
240 DATA 2A,5F,EE:    ' LD HL,(COMIEN)
250                   ' SIGUE:
260 DATA E5:          ' PUSH HL
270 DATA 11,64,EE:    ' LD DE,PRIMER
280 DATA 3A,63,EE:    ' LD A,(NUMCAR)
290 DATA 47:          ' LD B,A
300                   ' OTRO:
310 DATA 1A:          ' LD A,(DE)
320 DATA BE:          ' CP (HL)
330 DATA 23:          ' INC HL
340 DATA 28,03:       ' JR Z,IGUAL
350 DATA E1:          ' POP HL
360 DATA 18,10:       ' JR FIN
370                   ' IGUAL:
380 DATA 13:          ' INC DE
390 DATA 10,F5:       ' DJNZ OTRO
400 DATA E1:          ' POP HL
410 DATA CD,3A,EE:    ' CALL PRINT
420 DATA CD,9F,00:    ' CALL 009FH
430 DATA F6,20:       ' OR 20H
440 DATA FE,66:       ' CP 66H
450 DATA 28,0C:       ' JR Z,SALIR
460                   ' FIN:
470 DATA 23:          ' INC HL
480 DATA E5:          ' PUSH HL
490 DATA A7:          ' AND A
500 DATA ED,5B,61,EE:' LD DE,(FINAL)
510 DATA ED,52:       ' SBC HL,DE
520 DATA E1:          ' POP HL
530 DATA 38,D4:       ' JR C,SIGUE
540                   ' SALIR:
550 DATA 3E,0D:       ' LD A,0DH
560 DATA CD,A2,00:    ' CALL 00A2H
570 DATA 3E,0A:       ' LD A,0AH
580 DATA CD,A2,00:    ' CALL 00A2H
590 DATA C9:          ' RET
600                   ' PRINT:
610 DATA E5:          ' PUSH HL
620 DATA 7C:          ' LD A,H
630 DATA CD,4A,EE:    ' CALL HEXNUM
640 DATA 7D:          ' LD A,L
650 DATA CD,4A,EE:    ' CALL HEXNUM
660 DATA 3E,20:       ' LD A,20H
670 DATA CD,A2,00:    ' CALL 00A2H
680 DATA E1:          ' POP HL
690 DATA C9:          ' RET
700                   ' HEXNUM:
710 DATA F5:          ' PUSH AF
720 DATA 0F:          ' RRCA
730 DATA 0F:          ' RRCA
740 DATA 0F:          ' RRCA
750 DATA 0F:          ' RRCA
760 DATA CD,53,EE:    ' CALL HEX2
770 DATA F1:          ' POP AF
780                   ' HEX2:
790 DATA E6,0F:       ' AND 0FH
800 DATA C6,90:       ' ADD A,90H
810 DATA 27:          ' DAA
820 DATA CE,40:       ' ADC A,40H
830 DATA 27:          ' DAA
840 DATA CD,A2,00:    ' CALL 00A2H
850 DATA C9:          ' RET
860                   ' COMIEN:
870                   ' DEFS 2
880                   ' FINAL:
890                   ' DEFS 2:
900                   ' NUMCAR:
910                   ' DEFS 1:
920                   ' PRIMER:
930                   ' DEFS 255
```



era con el primer código de la secuencia, estas acciones son iguales, pero si vamos por, por ejemplo, el quinto código de la secuencia, es necesario recuperar dicho puntero del *stack*.

En el caso de que no se produjera la igualdad, tras incrementar el PUNTERO DE MEMORIA pasamos a verificar que no hemos alcanzado la última dirección a buscar. Si no es así, volvemos al comienzo, y en caso contrario hemos terminado la búsqueda.

Cuando se produzca la igualdad, tendremos que ver si el siguiente código en memoria coincide con el siguiente de la secuencia. Como ya hemos incrementado el PUNTERO DE MEMORIA, nos queda incrementar el PUNTERO DE CADENA y decrementar el contador. Como cada vez que un código consecutivo coincide con la secuencia decrementamos el contador, cuando hallemos la secuencia completa éste valdrá cero. Por tanto, si no es cero seguimos la búsqueda **con el siguiente código de la secuencia,** pero si es cero recuperamos del *stack* la dirección de memoria en que se encuentra el código que coincide con el primero de la secuencia.

Tras la impresión, llamamos a una rutina del sistema operativo (en la dirección 009FH) que espera a que pulsemos una tecla y nos devuelve en el acumulador el código ASCII correspondiente a la tecla pulsada. Si la tecla pulsada era la «F», salimos directamente de la rutina. Si no, incrementamos el PUNTERO DE MEMORIA y verificamos que no hayamos llegado al final.

Como podéis ver, el diagrama de flujo expresa ordenadamente este proceso. Cada toma de decisión (rombo) señala los caminos posibles con el SI y NO.

El listado BASIC adjunto se encarga de introducir en memoria los códigos correspondientes a la rutina ya ensamblada, si bien en sentencias *REM* indicamos los mnemónicos correspondientes. Una vez «pokeados» los códigos, el programa pide la longitud de la secuencia a buscar, la dirección de comienzo de búsqueda (en hexadecimal, con cuatro dígitos y sin el prefijo «&H»), la dirección de final de búsqueda (en hexadecimal, con cuatro dígitos y sin el prefijo «&H») y luego nos pide los códigos que forman la secuencia, «pokeando» todo esto en la memoria en los sitios indicados.

Rincón del lector

MAPA DE MEMORIA DEL SVI-328

Mi pregunta es si pueden publicar alguna lista, con el mapa de memoria del SVI-328 y/o algunas direcciones más que sean de utilidad.

Jorge Cortés Barea
Barcelona

Ultimamente, venimos publicando toda una serie de temas diversos, dentro de nuestra sección habitual dedicada al SVI-328.

En el número 19 (del mes de diciembre), se publicó el mapa de la ROM, los ports y las variables del sistema de este ordenador.

En el número 20 (del mes de enero), se inicia el análisis de los ports del ordenador y próximamente, nos dedicaremos a explicar detalladamente el resto de las características del ordenador.

COMPATIBILIDAD ENTRE MSX Y SVI-328

Desearía saber si con el adaptador de MSX para Spectravideo, también tendría la compatibilidad con los periféricos del sistema y si comentareis dicho periférico en la revista.

Abelardo Jiménez
Barcelona

Aún es pronto para determinar con certeza, si el adaptador de MSX de Spectravideo convierten al SVI-328 en MSX. Tenemos entendido que sólo hace compatible los programas en cartucho, manteniendo su propia configuración.

De cualquier manera, nos hemos puesto en contacto con CCG y pronto tendremos el adaptador para probarlo.

Si deseas más información sobre este adaptador, puedes ponerte en contacto con CCG, llamando al Tel.: (94) 462 46 52.

SOFTWARE PARA MSX

Me dirijo a Vds. para solicitarles la máxima información de que dispongan sobre el software MSX existente en el mercado.

También desearía me informasen sobre la diferencia entre el cartucho y el cassette.

Francisco Baeza Galindo
Almería

En el número 23 de MSX Magazine, que saldrá en el mes de abril, publicaremos un guía de programas, donde aparecerán todos los que hemos comentado a lo largo de todos estos meses. A su vez, para ese mismo número, tenemos pensado realizar una completa guia de libros, sin embargo, esta última opción está por determinar.

En la guía de software, haremos una breve introducción a todo lo que ha afectado el mercado de programas a lo largo de estos dos años, y explicaremos las diferencias fundamentales entre todos y cada uno de los soportes de los programas'

PROBLEMAS CON LA UNIDAD DE DISCOS

Poseo una unidad de discos Philips VY 0010 y varios diskettes. Todos, a excepción de uno de ellos, funcionan bien. Sin embargo, el defectuoso no lo puedo cargar de ninguna forma, ni desde el MSX-DISK BASIC, ni desde el MS-DOS. Al intentar la operación, la unidad de discos gira sin parar y finaliza con el mensaje de, Disk I/O Error. Mi pregunta es, ¿estará estropeado el diskette debido al mal uso o por el contrario será debido al uso incorrecto de alguna instrucción? ¿Podré solucionarlo sin perder la información que en él tengo almacenada? ¿Tendré que formatearlo?

Crisendo García González
Santander



Desde luego, si de todos los discos que posees sólo hay uno que no carga, automáticamente habrá que descartar el uso indebido de cualquier instrucción puesto, que como me indicas, el resto de los discos cargan bien. Principalmente, creemos que se deba a que haya un defecto en el disco, lo que imposibilita su lectura. Este defecto puede ser debido a cualquier tipo de suciedad, polvo, rayado, etc., lo que hará el disco difícil de leer aún por cualquier programa que lea sectores.

Sin embargo, como suele decirse, «la esperanza es lo último que se pierde». En el número 17 de MSX Magazine (del mes de octubre), en la sección de Trucos (pág. 65) publicamos un corto programa que permite recuperar la información de un disco defectuoso. Es una buena ocasión para probarlo ¿no crees?

De cualquier manera, siempre queda el último recurso, formatear el disco. Sin embargo, esto significa que perderás toda la información que tengas grabada en él.